



Lez03a - Elis - IT Architect



Setup IDE
Ing. Alberto Bucciero

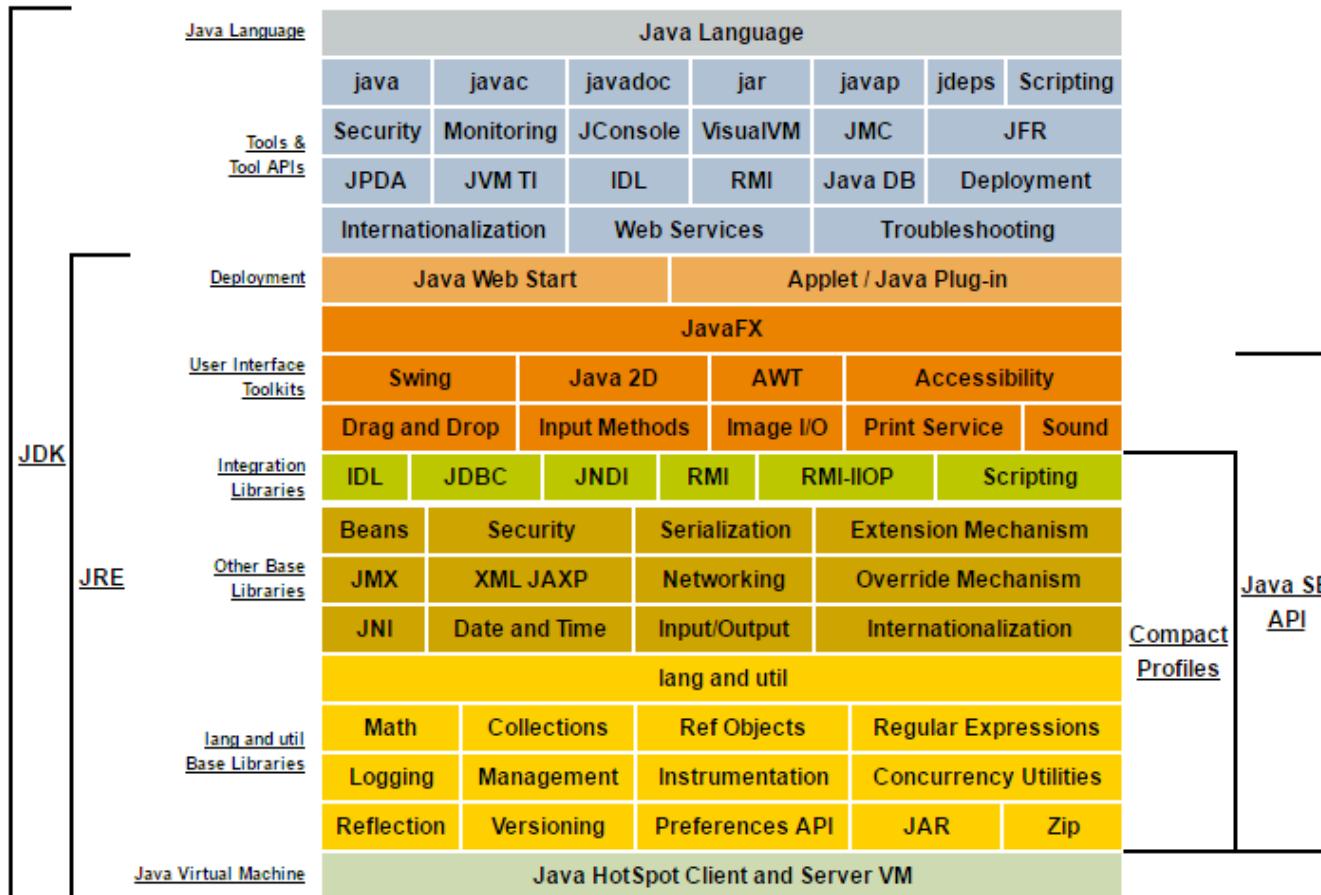
- Setup ambiente di sviluppo
 - Prerequisiti : JDK
 - Installazione JEE
 - Eventualmente IDE (Netbeans o Eclipse)

JDK

- URL:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Versione più aggiornata:
Java SE Development Kit 8u112
- NB: è un eseguibile → dipende dal SO

JDK

Description of Java Conceptual Diagram



Installazione di Java

1. Download dell'ultima versione della Piattaforma Java (JDK) dal sito:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> [versione più aggiornata Java SE Development Kit 8u112]
 - Attenzione: scegliere la versione JDK (che contiene il compilatore e l'ambiente di esecuzione) e non la versione JRE (che contiene solo l'ambiente di esecuzione).
 - Verranno proposte due versioni tra cui scegliere:
 - Windows x86
 - Windows x64
 - La versione x64 (64 Bit) funziona solo su computer abbastanza recenti (pochi anni).
 - Molto probabilmente funzionerà sulla grande maggioranza dei vostri computer. La versione x86 (32 Bit) funziona su tutti i computer (nuovi e vecchi), ma potrebbe essere leggermente meno performante della versione x64.
2. Installazione del JDK eseguendo il file scaricato dal sito. L'installazione dovrebbe essere abbastanza automatica...

Modifica della variabile di sistema PATH.

- Al fine di poter eseguire il compilatore Java e la Java Virtual Machine “manualmente” tramite il prompt dei comandi è necessario modificare la variabile di sistema PATH eseguendo le seguenti operazioni:
 - Identificare la directory in cui è stato installato il JDK. In particolare, è necessario conoscere il nome della directory che contiene il file “javac.exe”. Molto probabilmente il percorso di tale directory sarà simile a: C:\Program Files\Java\jdk1.7.0_40\bin
 - All’interno del menu start fare click con il tasto destro su “Computer” e selezionare “Proprietà” (in alternativa, “Pannello di Controllo”->“Sistema e sicurezza”->“sistema”)
 - Selezionare “Impostazioni di sistema avanzate”

- Selezionare “Variabili d’ambiente”
- Tra le “Variabili di sistema” (la seconda lista) individuare “Path” e selezionare la stessa tramite un doppio click
- Il campo “Valore variabile” contiene una lunga lista di directory separate da ; Aggiungere in fondo a tale lista un nuovo ; e la directory identificata in precedenza

- Per verificare che tutto funzioni aprire il prompt dei comandi (“Menù start”->“tutti i programmi”- >“Accessori”->“Prompt dei comandi”) e digitare il comando:

```
java –version
```

il sistema dovrebbe rispondere con tre righe simili a: java version "1.7.0_40" Java(TM) SE Runtime Environment (build 1.7.0_40-b43) Java HotSpot(TM) Client VM (build 24.0-b56, mixed mode, sharing)

```
javac –version
```

- il sistema dovrebbe rispondere con: javac 1.8.0.112

Eclipse - releases

Table 1. Eclipse releases

Release	Rename name	Release year
4.2	Juno	2012
4.3	Kepler	2013
4.4	Luna	2014
4.5	Mars	2015
4.6	Neon	2016
4.7	Oxygen	2017

Installazione di Eclipse

- Download dell'ultima versione di Eclipse dal sito: <http://www.eclipse.org/downloads/>
- Tra le versioni proposte, l'edizione Neon 4.6 è la più aggiornata.
- Come nel caso di Java, la versione 64 Bit dovrebbe funzionare sulla maggior parte dei computer (soprattutto se recenti). La versione 32 Bit invece funziona sicuramente su tutti i computer, ma potrebbe essere leggermente meno performante.
- Il programma Eclipse è contenuto in un unico file eseguibile che ha un nome simile a: `eclipse-inst-win64.exe`
- Scaricare ed eseguire il file
- Scegliere “crea collegamento” accettando di posizionare il collegamento sul Desktop. A questo punto dovrebbe essere comparsa l’icona viola del programma Eclipse sul desktop, che consente di eseguire il programma stesso.

The Eclipse Neon (4.6) release requires Java 8 to run.



Please note that you need a 64 JVM to run a 64 bit Eclipse and a 32 bit JVM to run a 32 bit Eclipse.

Eclipse

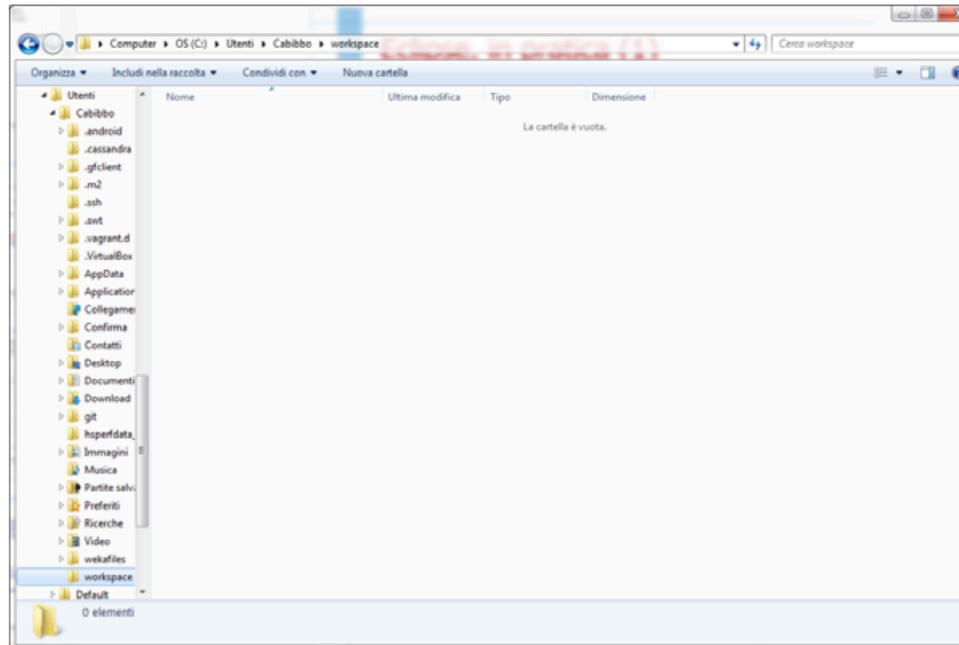
- Eclipse è un ambiente integrato di sviluppo
 - inizialmente sviluppato da IBM
 - poi rilasciato come progetto open-source
 - ora è una comunità open-source, il cui obiettivo è lo sviluppo di una piattaforma di sviluppo aperta composta da una famiglia di strumenti estensibili orientati allo sviluppo del software
 - disponibile per numerose piattaforme – Windows, Linux, Mac
 - sul sito web www.eclipse.org
- L'uso di Eclipse può risultare inizialmente poco immediato ma dopo un po' di pratica è molto comodo ed efficace

Due nozioni di Eclipse

- Workspace
 - letteralmente, “spazio di lavoro”
 - l’obiettivo di un workspace è di memorizzare tutto il codice relativo a un programma o sistema software (anche grande)
 - in pratica, un workspace corrisponde a una cartella sul file system – e tutte le cartelle e i file in esso contenuti
- Project
 - letteralmente, “progetto”
 - l’idea è che un grande sistema software non è realizzato come un singolo progetto – ma piuttosto come un insieme di progetti correlati
 - un project rappresenta dunque una porzione di un sistema software, ovvero una porzione di workspace
 - in pratica, un insieme di classi

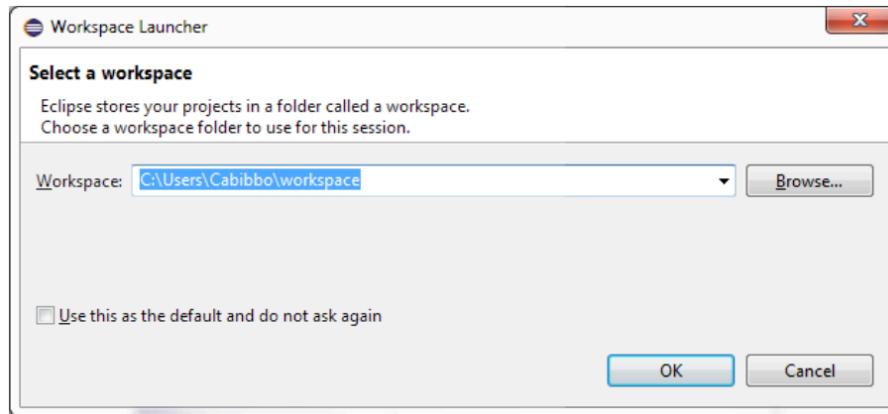
Eclipse, in pratica

- Per prima cosa, è utile creare sul disco una cartella per memorizzare un workspace
 - si suggerisce di dedicare un workspace all'intero corso
 - il default di Eclipse è una cartella workspace – nella propria cartella Documenti



Eclipse, in pratica

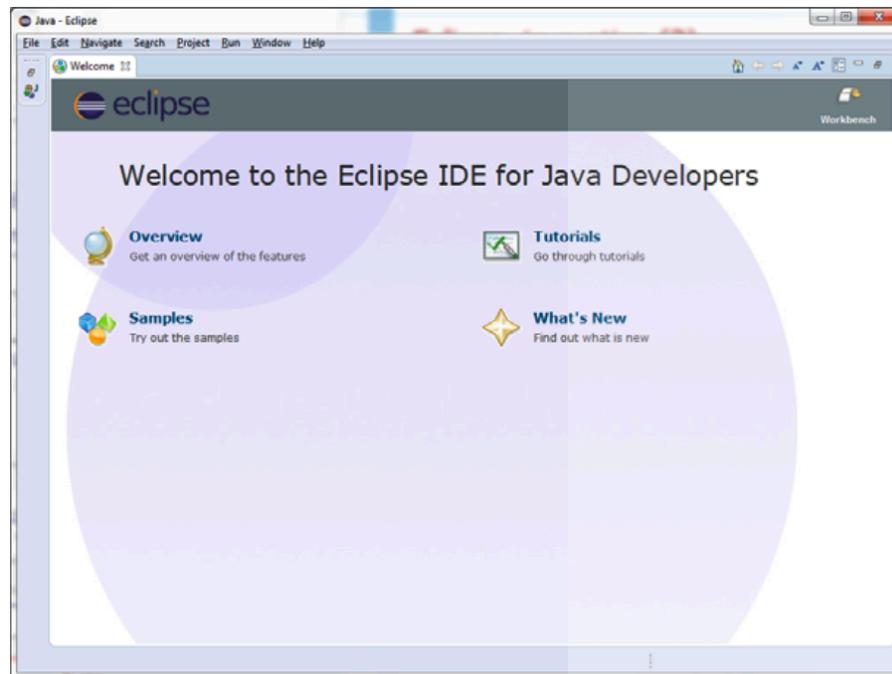
- All'avvio di Eclipse viene richiesto di selezionare un workspace
 - è possibile selezionare la cartella **workspace** nella nostra cartella **Documenti**



- le volte successive Eclipse si ricorderà della nostra scelta e ce la riproporrà come opzione iniziale – dovremo semplicemente confermarla

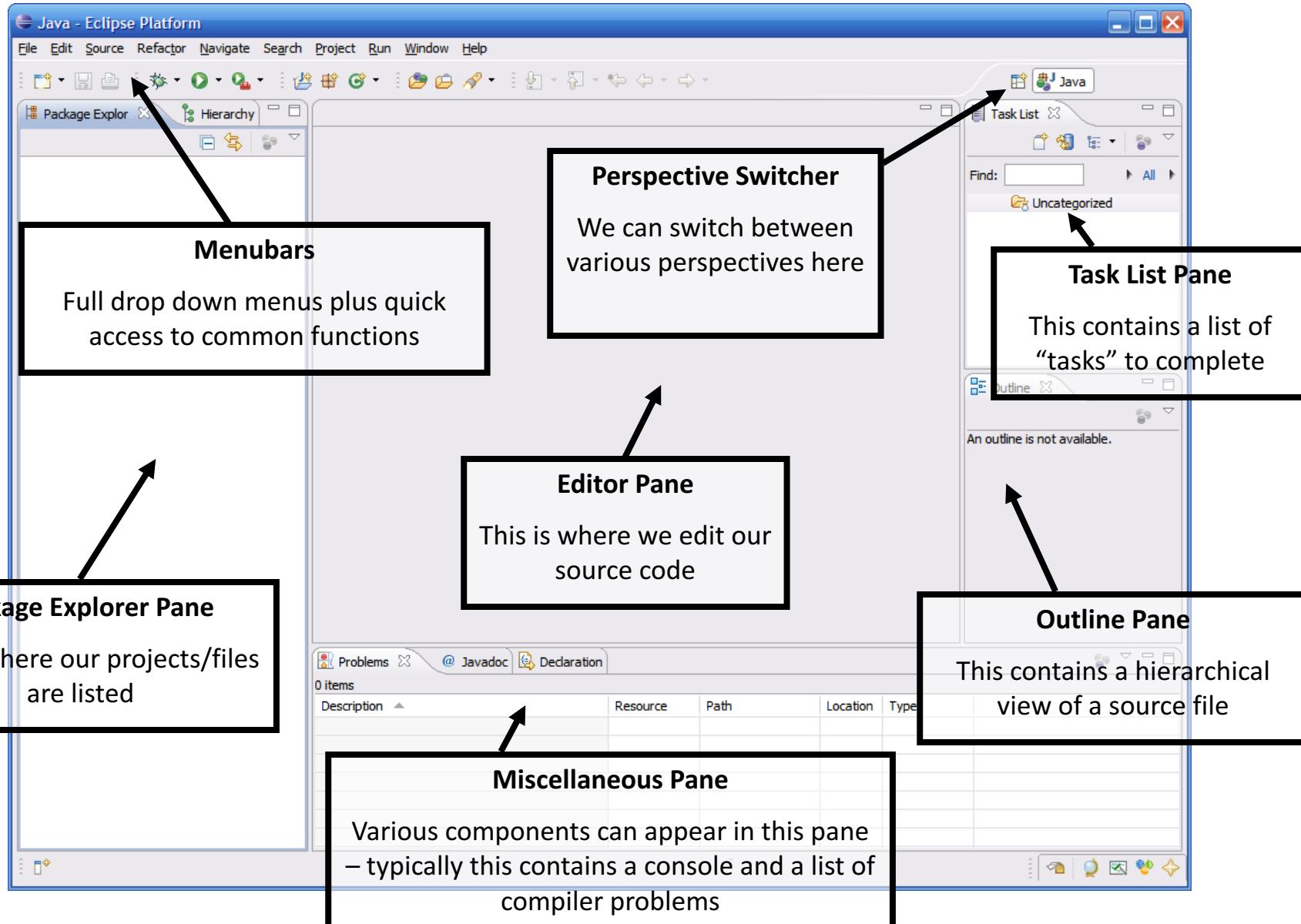
Eclipse, in pratica

- La prima volta che viene utilizzato un workspace, viene proposta una schermata di benvenuto “Welcome” – che possiamo chiudere
 - intervenendo sulla crocetta sulla linguetta “Welcome”, non sulla crocetta per chiudere la finestra



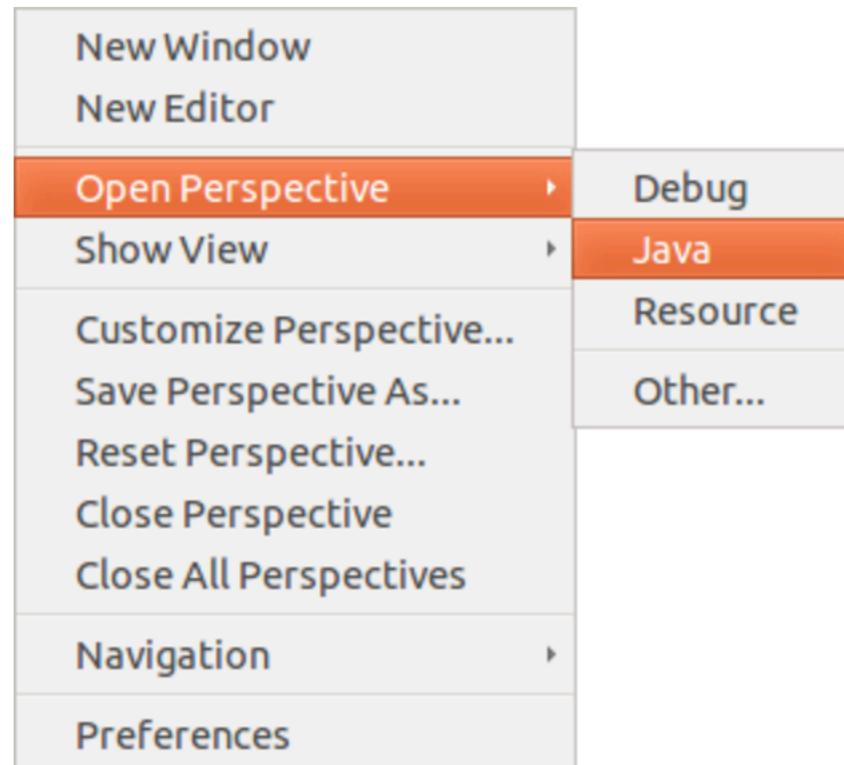


Eclipse IDE Components



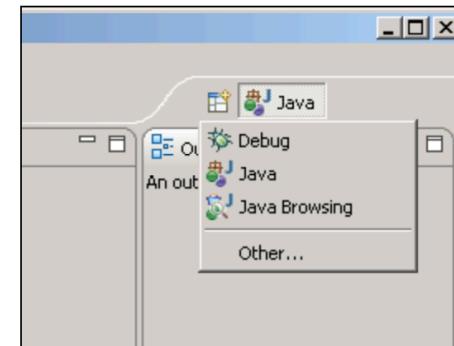
Perspectives in Eclipse

- Eclipse provides different *perspectives* for different tasks. The available *perspectives* depend on your installation. For Java development you usually use the *Java Perspective*, but Eclipse has much more predefined *perspectives*, e.g., the *Debug perspective*.



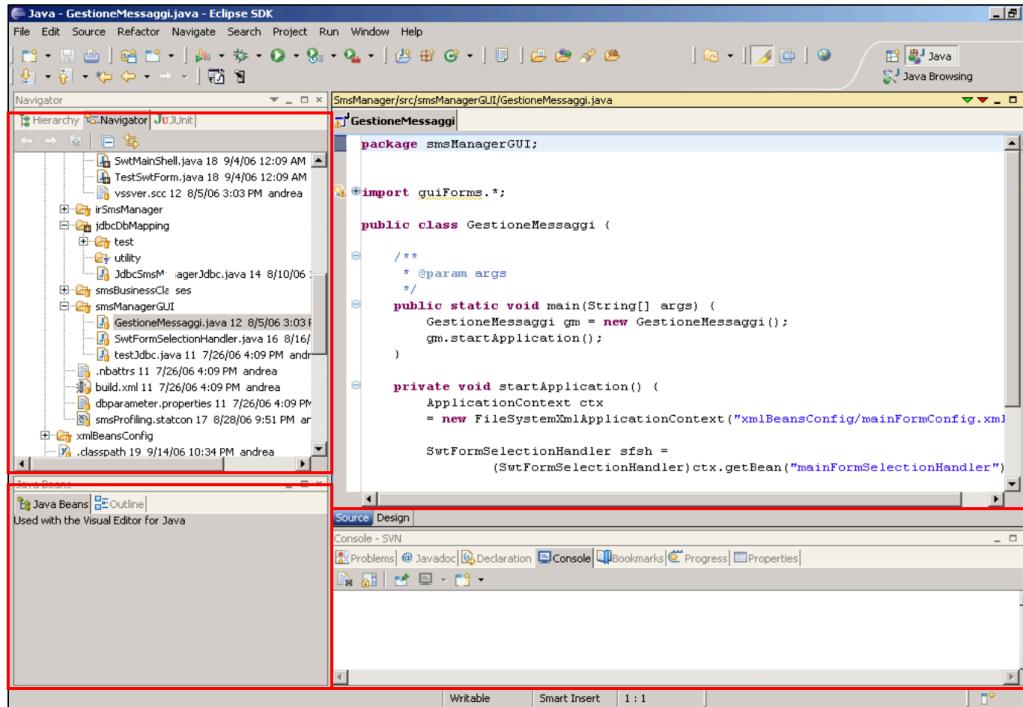
Le perspectives

- L'interfaccia grafica di Eclipse è organizzata a perspectives. Le perspectives non sono altro che dei raggruppamenti di funzionalità dell'IDE, fatti in base ad una specifica operazione di sviluppo. La Java perspective ad esempio riunisce strumenti di stesura e organizzazione del codice mentre la Debug perspective fornisce strumenti in fase di debug.



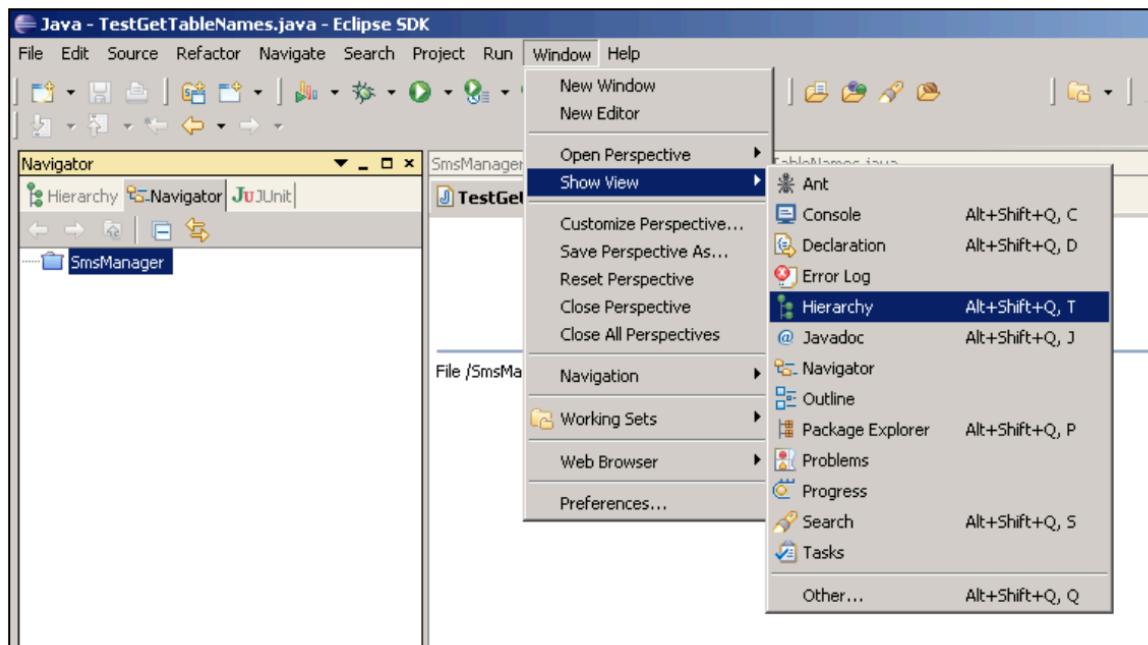
Le viste (1/3)

- Le viste sono “riquadri” organizzati in schede che offrono supporto per organizzare e stendere il codice. Ci sono tre gruppi schede che possono ospitare le viste:



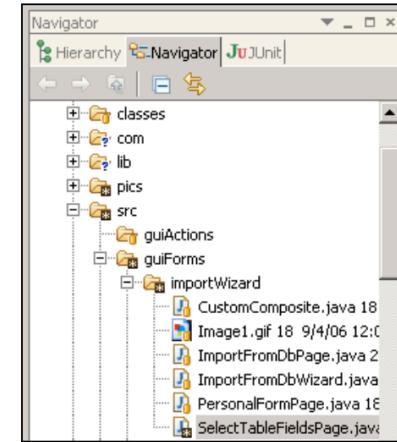
Le viste (2/3)

- Le varie viste possono essere spostate da un gruppo all'altro mediante semplice trascinamento della scheda sul nuovo gruppo.
- Le viste sono molto numerose. Per aggiungerne di nuove occorre accedere al menù "Window ☰ Show Views".

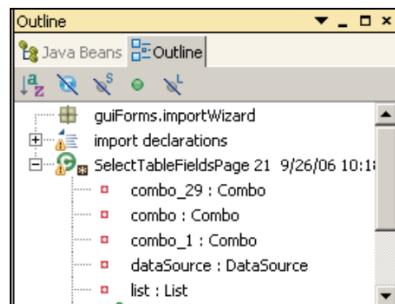


Le viste (3/3)

- Tra le viste più utili c“è la “Navigator”:



- Anche la vista “Outline” risulta molto comoda:

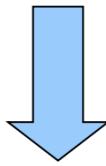


Tasti di scelta rapida (1/2)

- Sono il vero punto di forza dell'ambiente e richiamano funzioni sofisticate dell'editor. Sono diverse decine ma vale la pena tenerne a mente qualcuno...
- Ctrl + Space: il più “gettonato” in assoluto. Attiva l'intellisense in ogni punto del codice, completando i nomi delle variabili o parole chiave. Utilissimo dopo le new...

Esempio:

- . `StringBuffer sb = new`

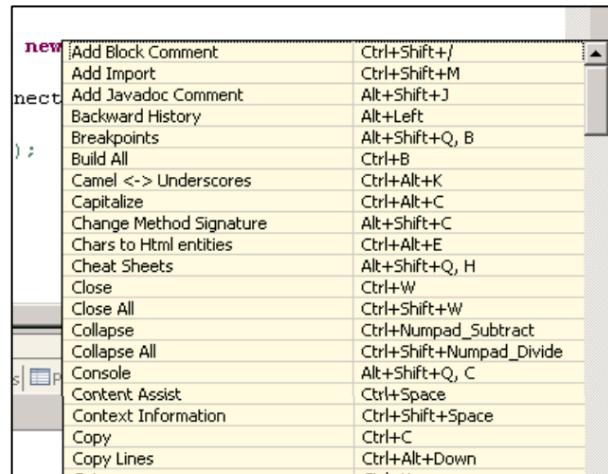


(premendo Ctrl + Space)

- . `StringBuffer sb = new StringBuffer`

Tasti di scelta rapida (2/2)

- Ctrl + Shift + Space: in prossimità di un metodo restituisce informazioni sui parametri di invocazione
- Ctrl + Shift + X: converte il testo selezionato in maiuscolo. Con la Y al posto della X il testo è convertito in minuscolo.
- Ctrl + Shift + L: il più importante. Richiama la lista con tutte le combinazioni dei tasti di scelta rapida:

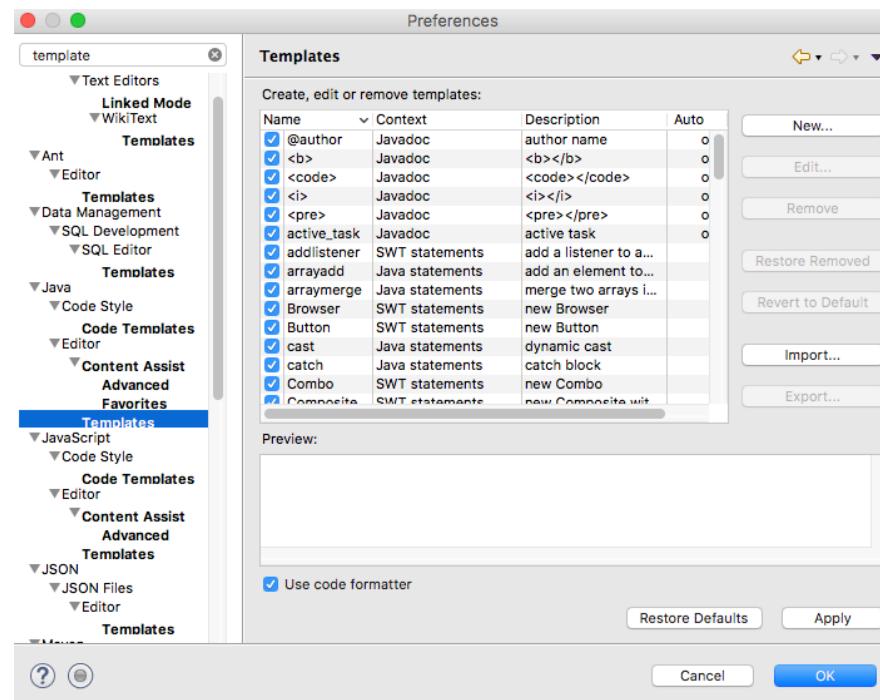


Template (1/2)

- I template sono modelli di codice che ricorrono nell“uso di certi costrutti sintattici di Java. Tramite I template quando questi costrutti si presentano possiamo fare in modo che Eclipse li costruisca in automatico risparmiando tempo (ed errori!).
- Es: in Eclipse se voglio usare un ciclo for each è sufficiente scrivere foreach e premere ctrl + space. Eclipse si occuperà di creare un modello del costrutto for each al quale andremo ad aggiungere solo la variabile di iterazione e la collezione di oggetti da iterare.
- Es: scrivo if premo ctrl + space: mi ritrovo il costrutto if pronto all“uso.
- I template si possono creare e modificare dalla voce “Window ☰ Preference”.

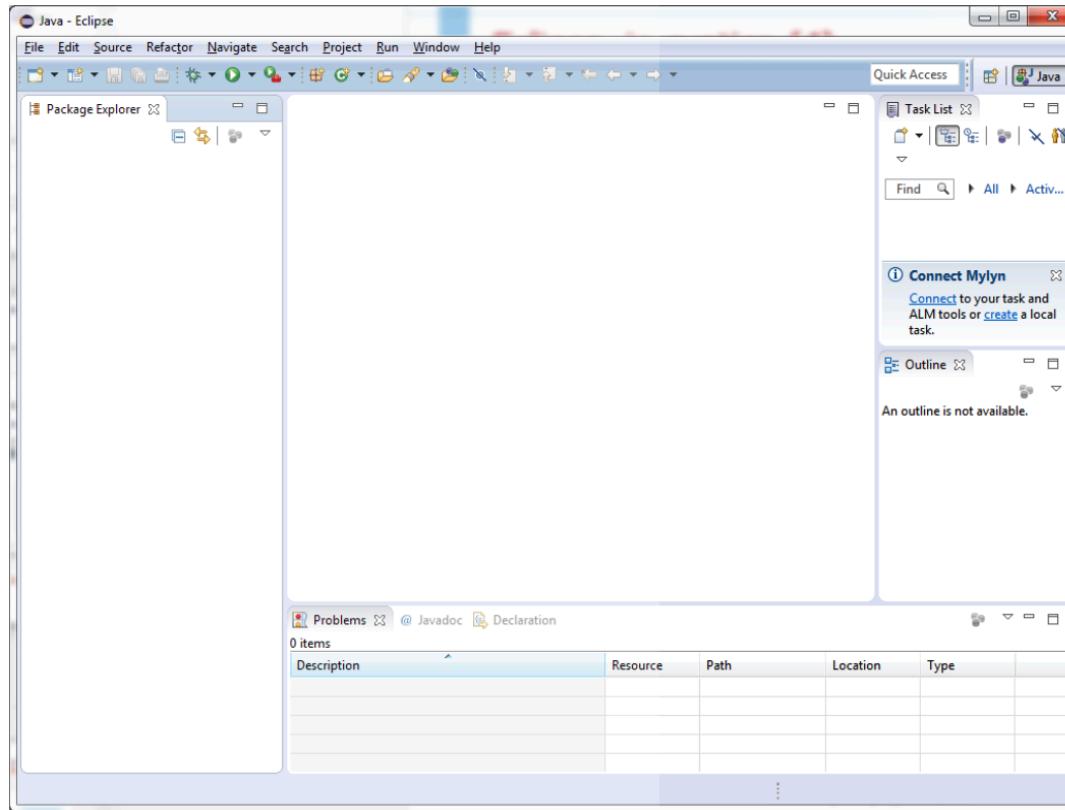
Template (2/2)

- IMPORTANTE: In molti menù di Eclipse abbiamo la possibilità di trovare la voce che vogliamo configurare facendo una ricerca testuale sul suo nome!



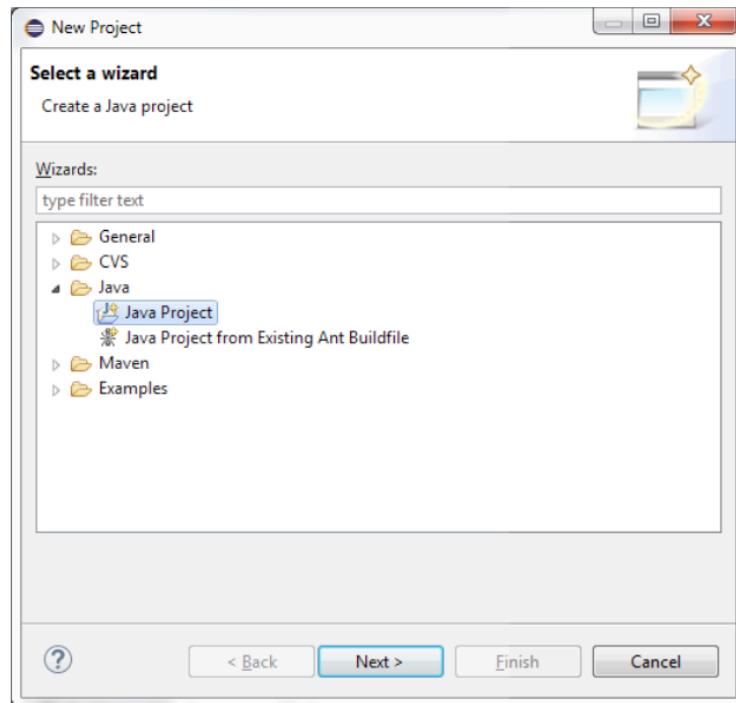
Eclipse, in pratica

- Ora abbiamo di fronte questa schermata



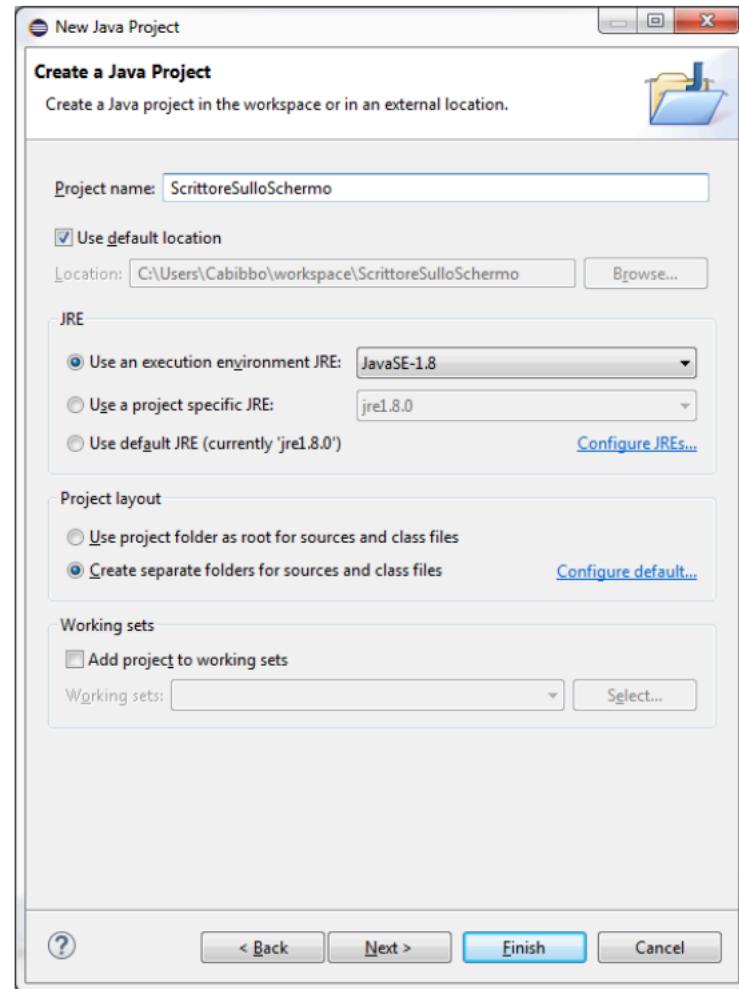
Eclipse, in pratica

- E' necessario creare (oppure aprire) un progetto
 - ad esempio, definendo un progetto per ciascuna applicazione – oppure un progetto per ciascun diverso argomento del corso
 - dal menu File -> New -> Project..., scegliendo poi Java -> Java Project



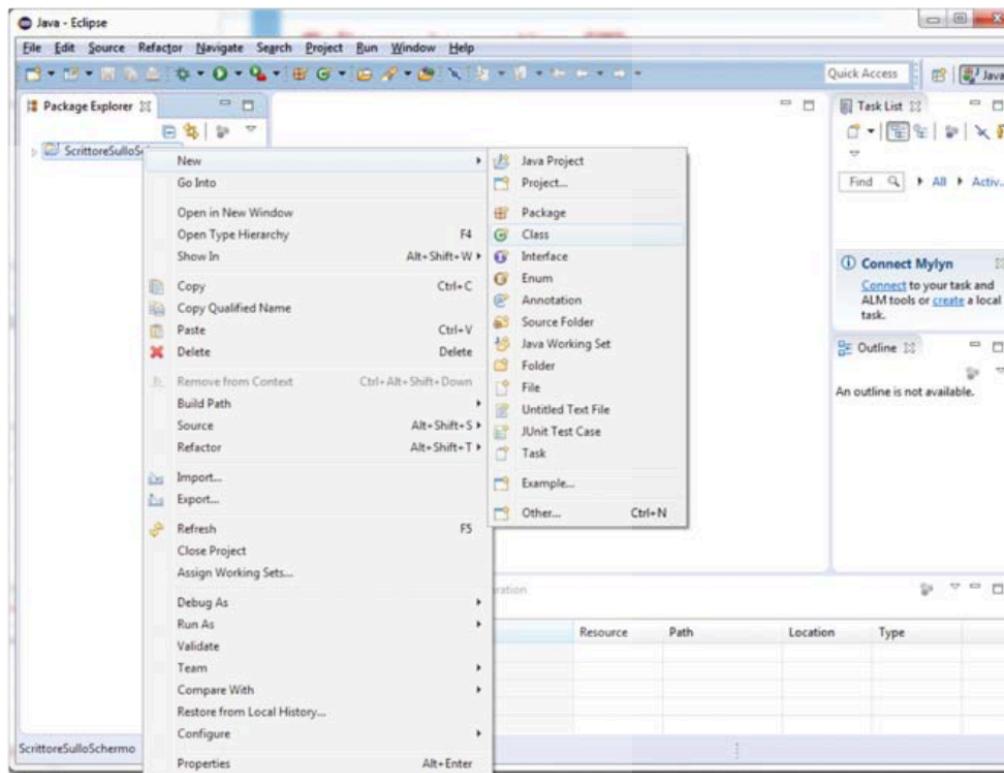
Eclipse, in pratica

- Dobbiamo scegliere un nome per il nostro progetto
 - in un workspace, ogni progetto deve avere un nome distinto
 - ad esempio,
ScrittoreSulloSchermo



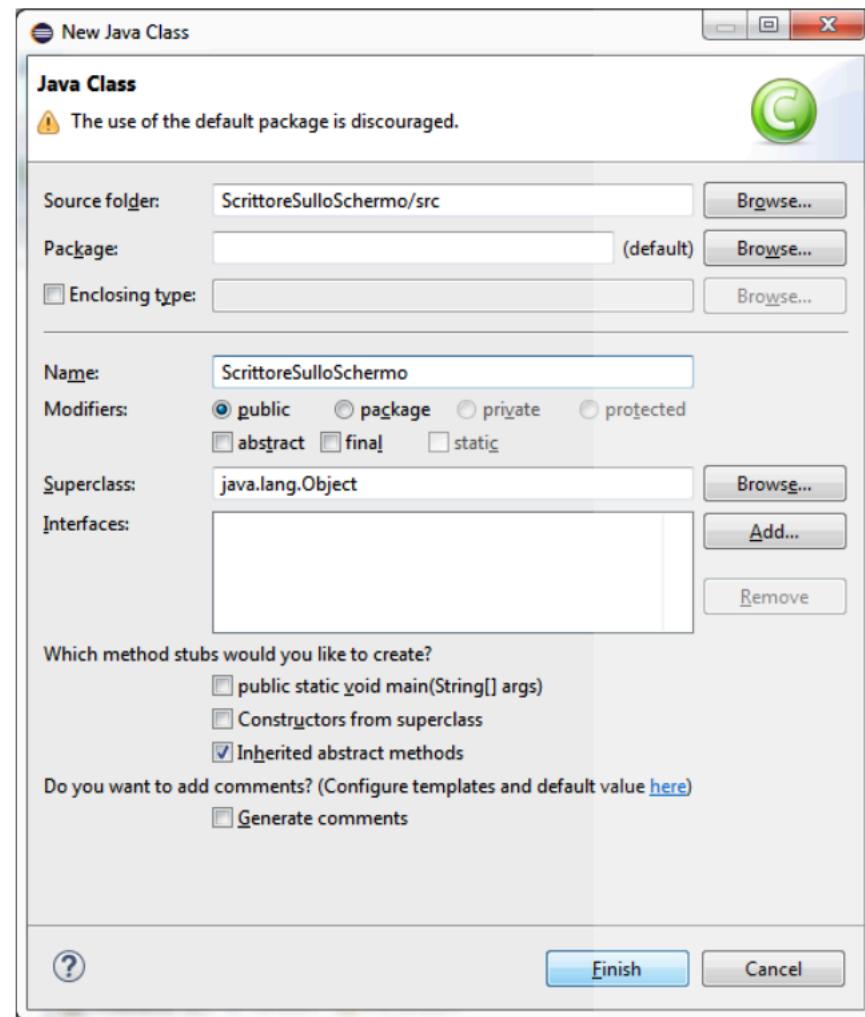
Eclipse, in pratica

- Il prossimo passo è definire una classe
 - dal menu File -> New -> Class – oppure dal menu contestuale (click con il tasto DESTRO del mouse) sul nome del progetto

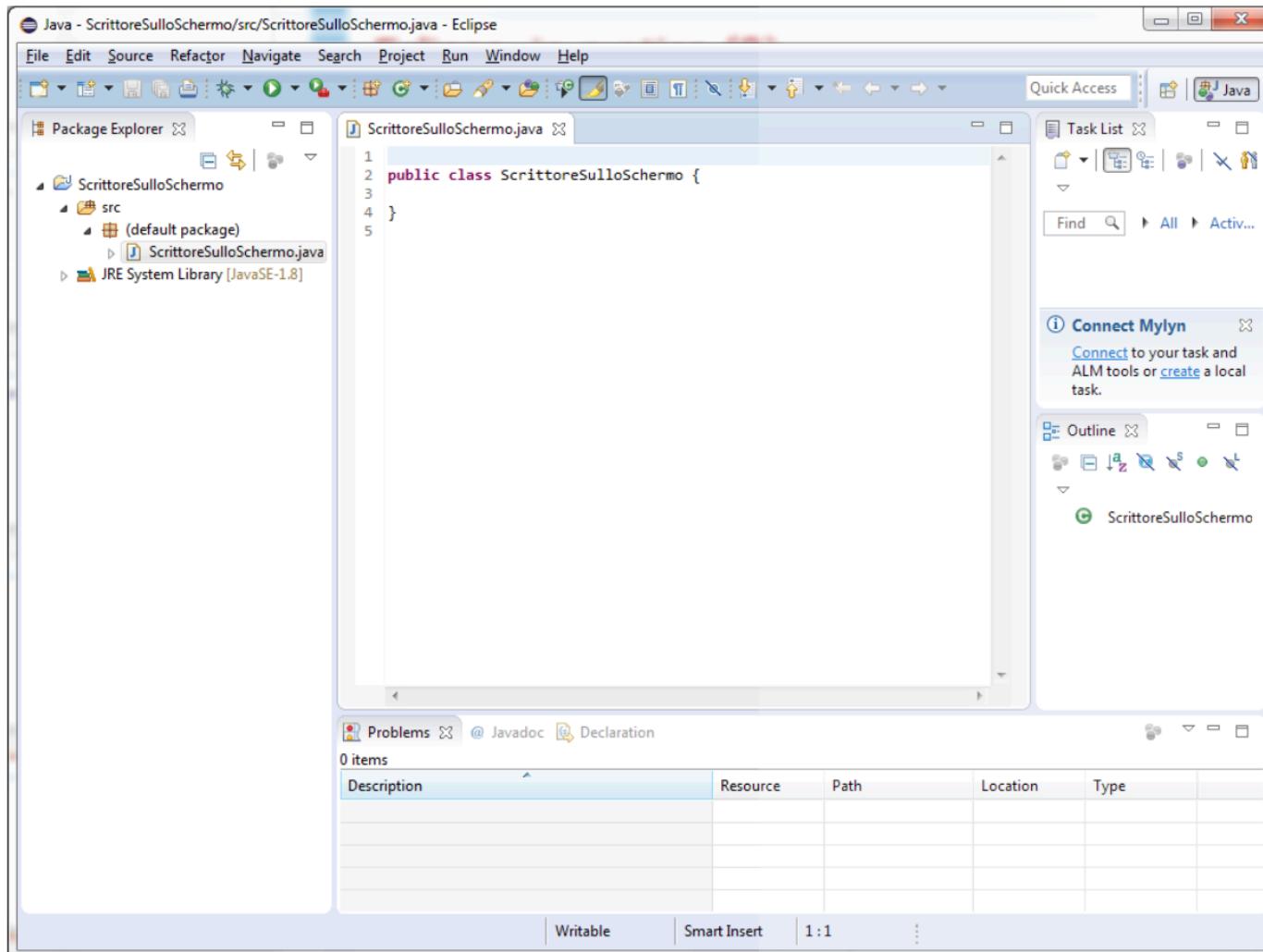


Eclipse, in pratica

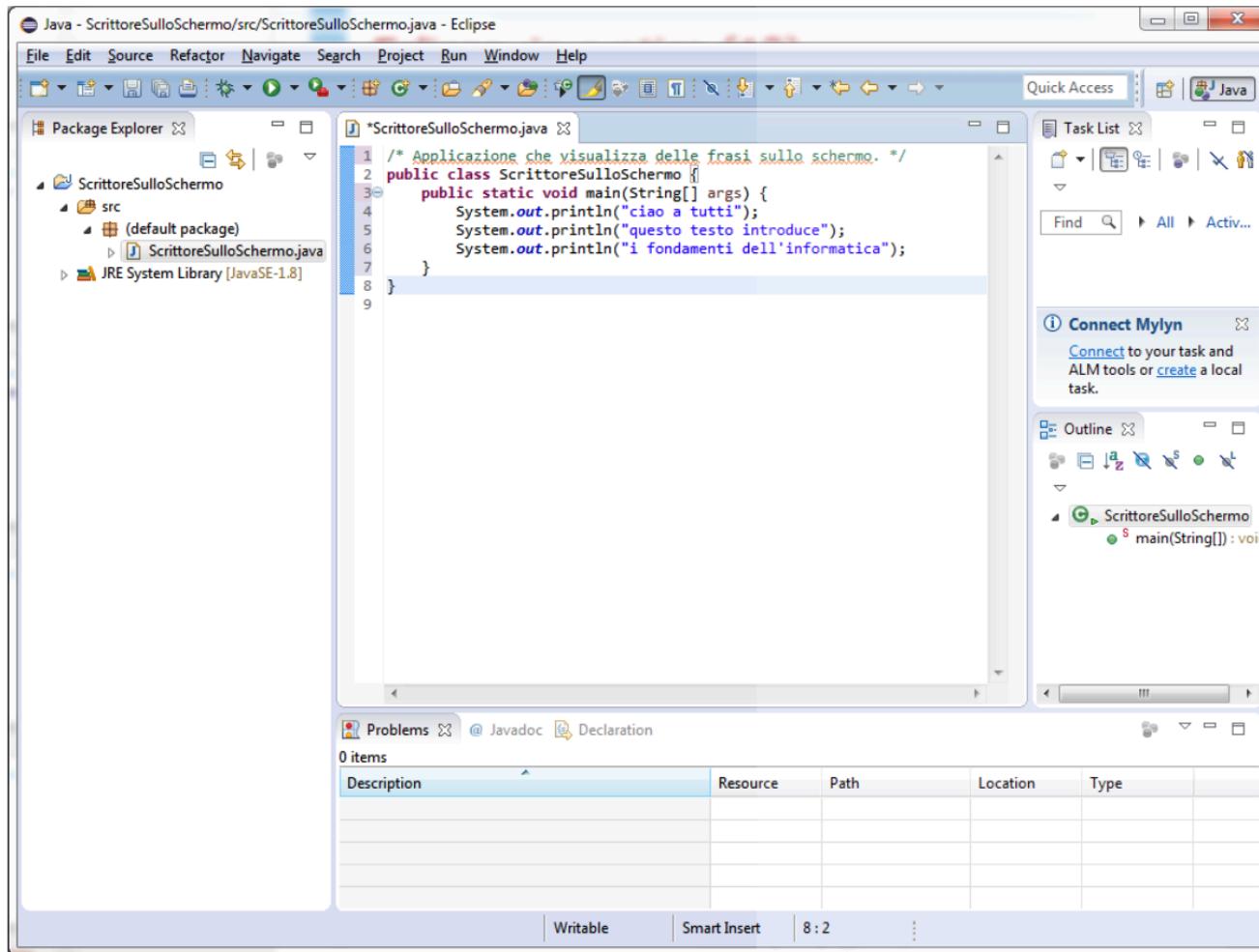
- Ogni classe in un progetto deve avere un nome distinto
 - ad esempio,
ScrittoreSulloSchermo



Eclipse, in pratica



Eclipse, in pratica



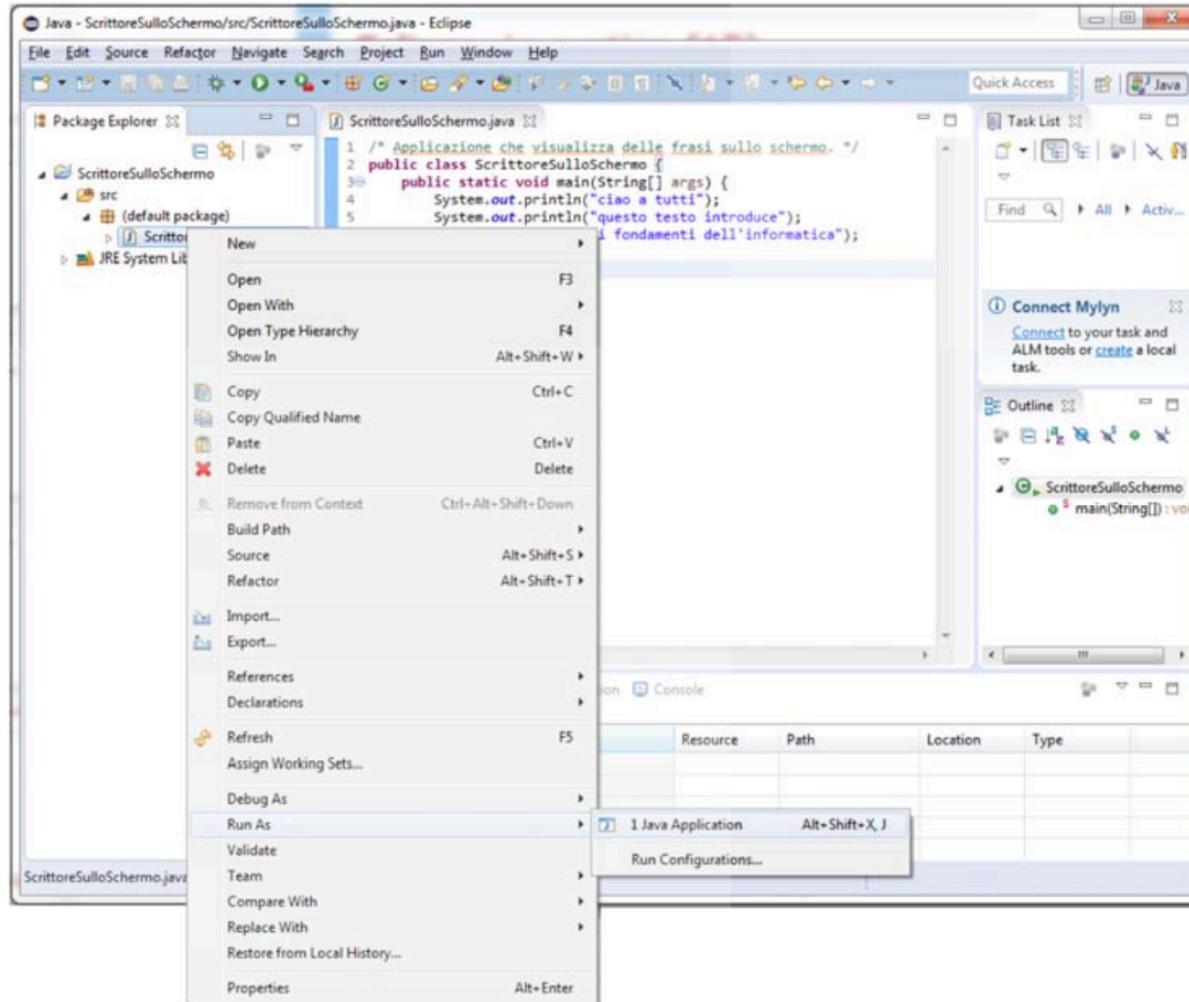
Eclipse, in pratica

- A differenza di altri IDE, nella sua configurazione iniziale Eclipse prova a compilare – automaticamente e continuamente – tutto il codice che scriviamo, appena lo scriviamo
 - non è necessario compilare il programma in modo esplicito
 - ma non è vero che non avviene nessuna compilazione
- Attenzione, il compilatore non è Eclipse
 - sul computer deve essere installato anche Java SDK
 - Eclipse richiede l'esecuzione di javac per noi

Eclipse, in pratica

- Per eseguire un programma
 - si clicca con il tasto DESTRO del mouse – non il sinistro – sul nome del file per la classe applicazione
 - si seleziona la voce Run As -> Java Application
 - il risultato dell'esecuzione sarà mostrato nella finestra “Console in basso a destra

Eclipse, in pratica



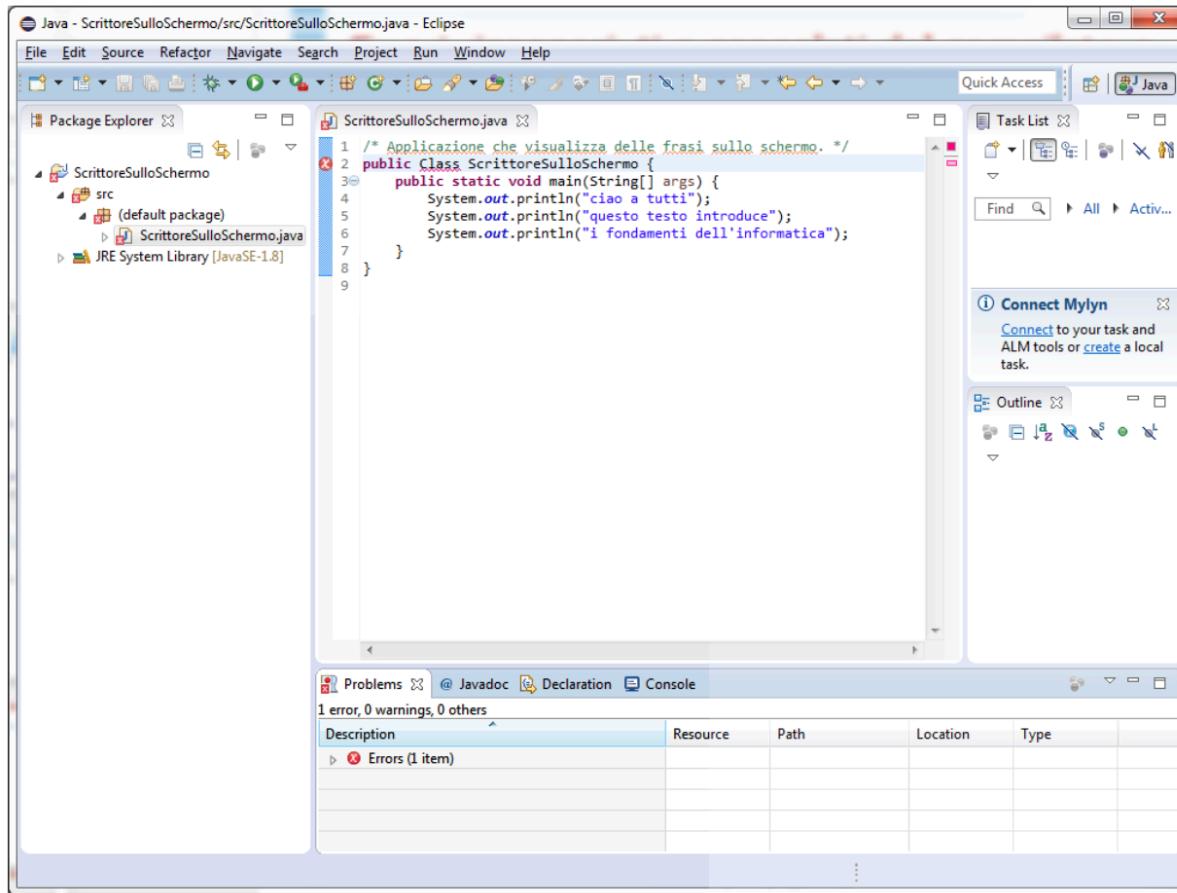
Eclipse – in sintesi

- Workspace
 - ne devo creare almeno uno
 - uno, una volta per tutte – oppure
 - uno per ciascun sistema da realizzare
- Project
 - ne devo creare almeno uno
 - uno per ciascuna argomento del corso – oppure
 - uno per ciascuna applicazione

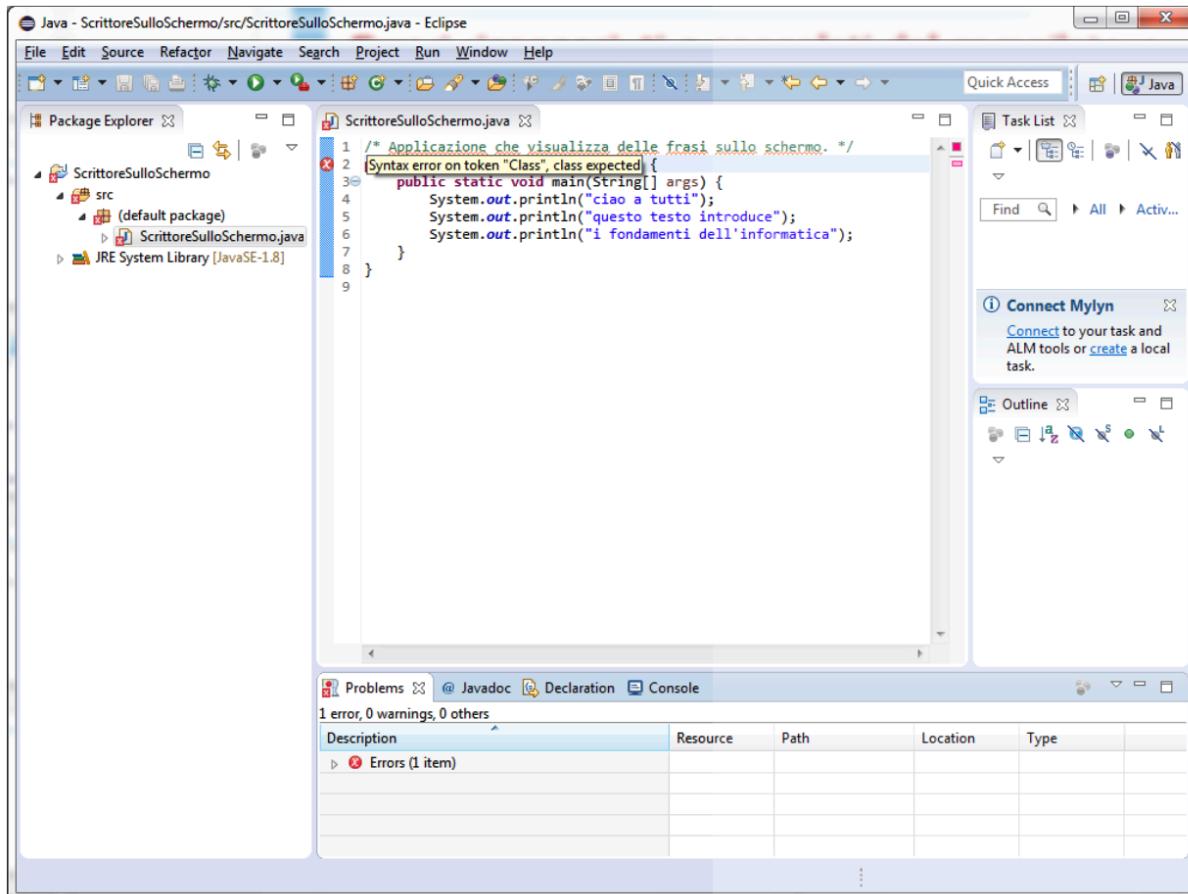
Eclipse – in sintesi

- Per scrivere un'applicazione
 - devo aggiungere – e poi scrivere – tutte le classi che servono in particolare, almeno una deve essere una classe applicazione
 - nella creazione della classe corrispondente, si può sfruttare un'opzione per creare automaticamente un metodo main inizialmente vuoto
 - Which method stubs would you like to create? public static void main(String[] args)
 - la compilazione delle classi è “automatica”
 - va richiesta l'esecuzione della “classe applicazione”

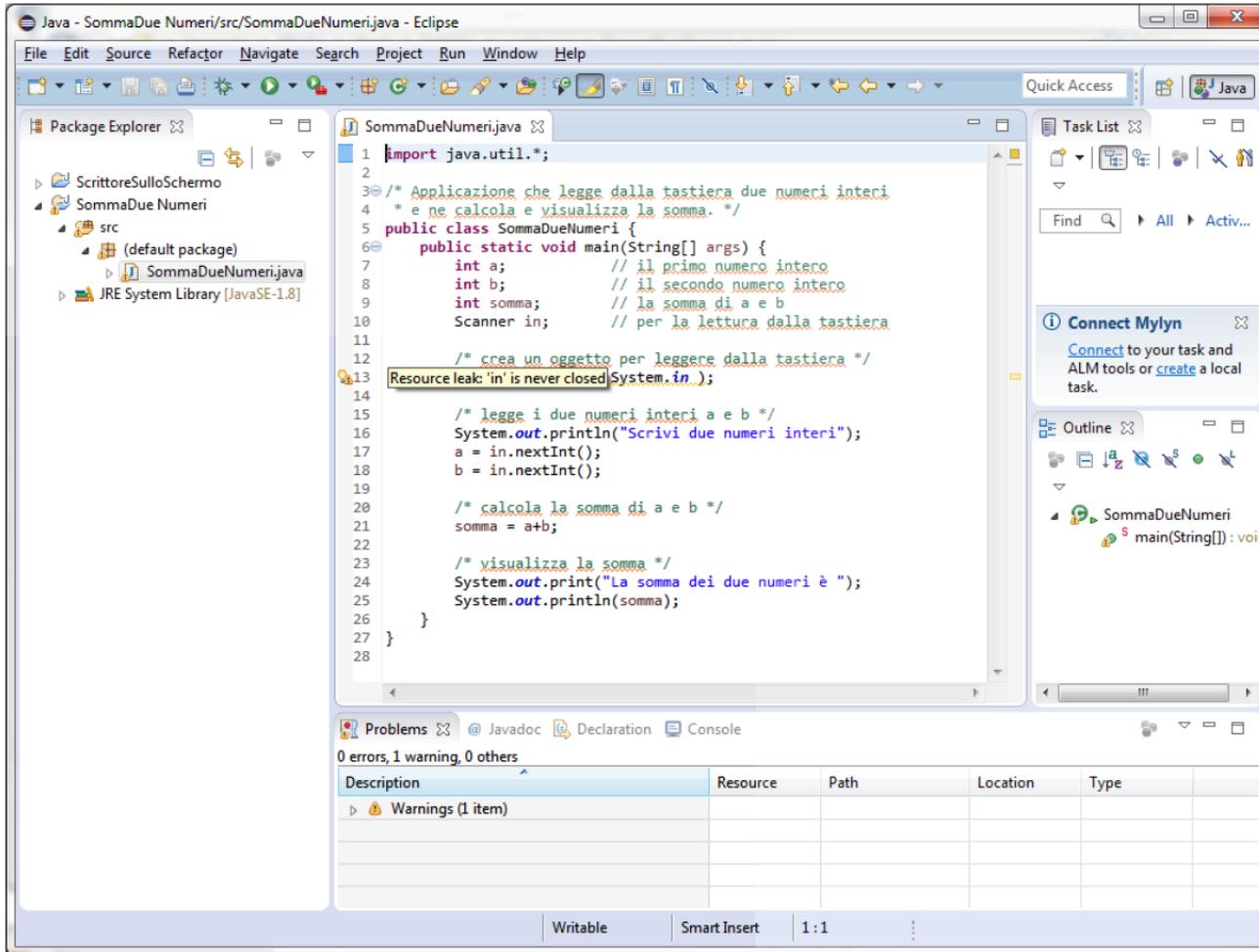
Segnalazione di errori



Segnalazione di errori



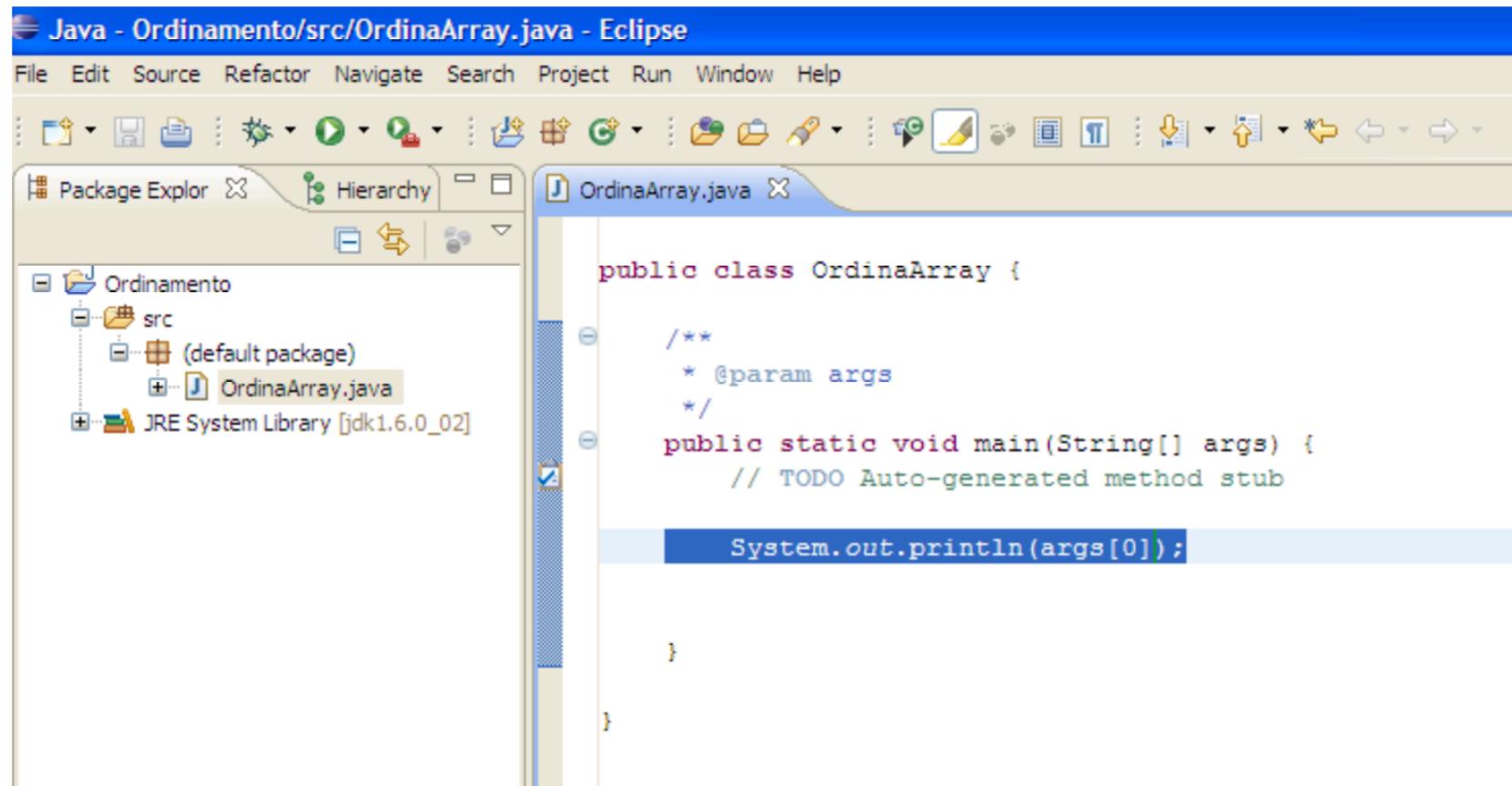
Warning (avvertimenti)



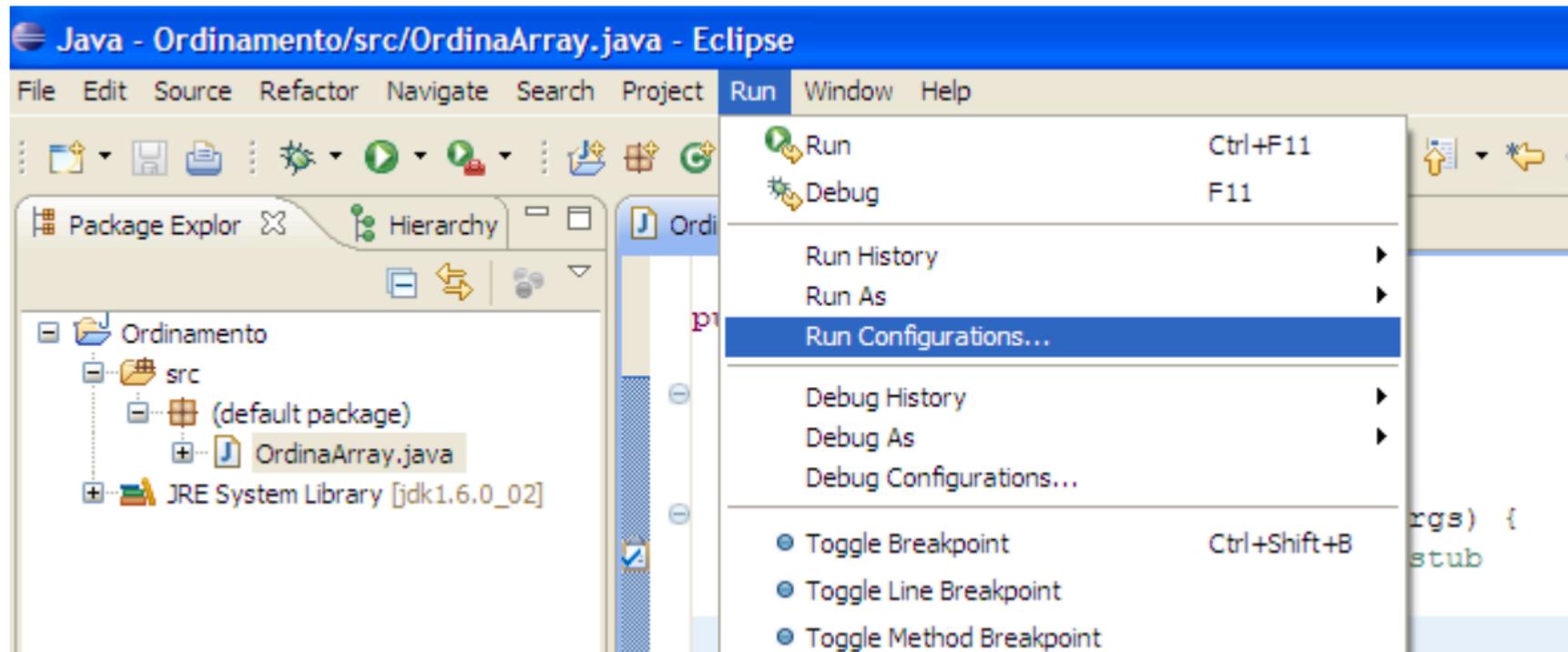
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project "SommaDueNumeri" with its source files.
- SommaDueNumeri.java Editor:** Displays Java code for summing two integers. A warning is highlighted at line 13: `Resource leak: 'in' is never closed [System.in];`
- Problems View:** Shows 0 errors, 1 warning, and 0 others.
- Outline View:** Shows the class structure.
- Task List View:** Shows a task titled "Connect Mylyn".

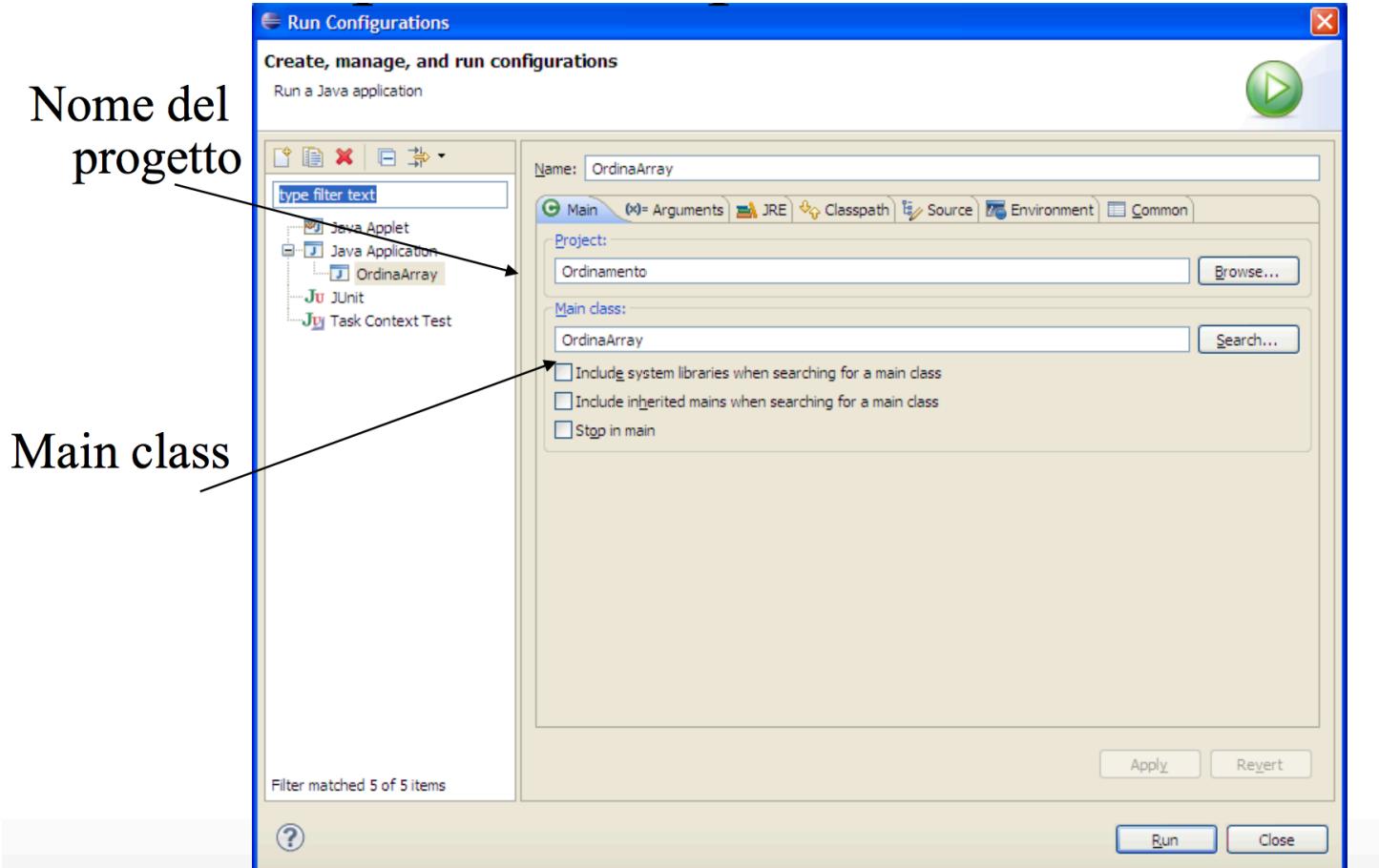
Eclipse: eseguire una applicazione e specificare parametri



Eclipse: eseguire una applicazione e specificare parametri

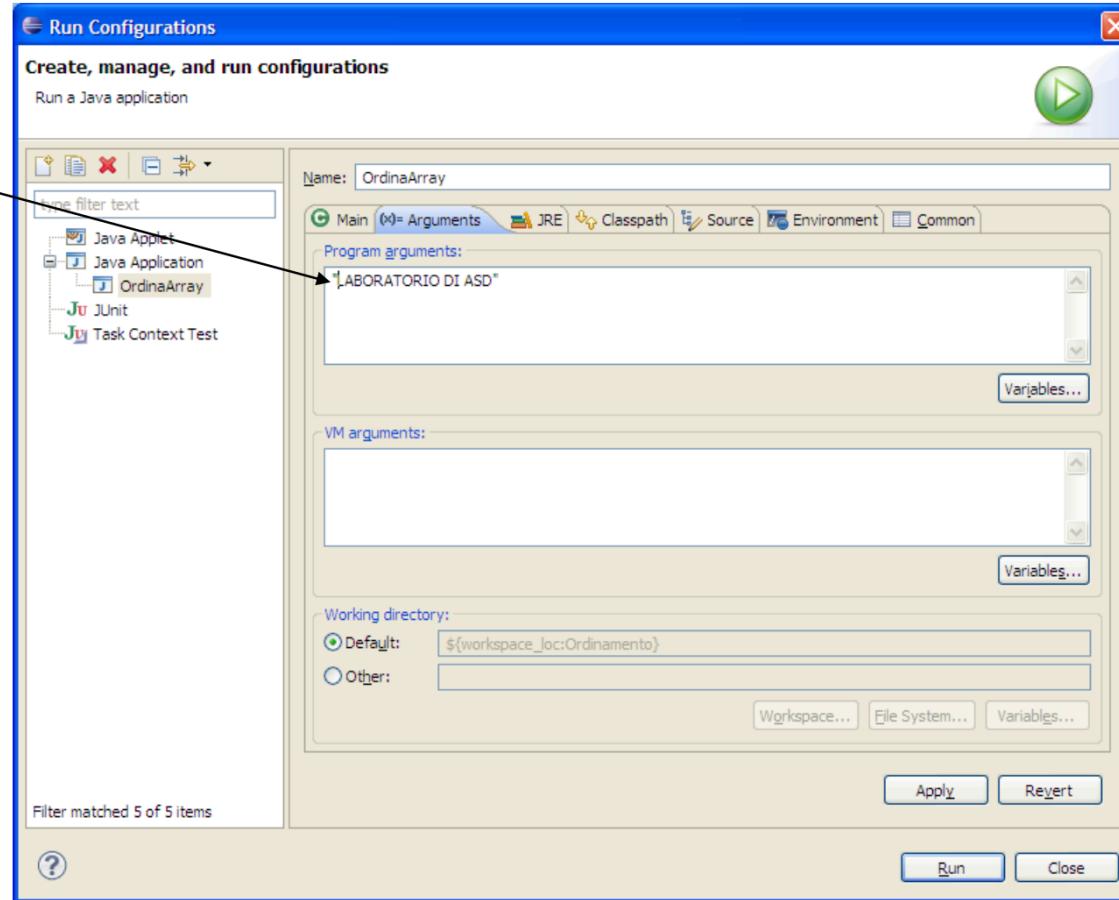


Eclipse: eseguire una applicazione e specificare parametri



Eclipse: eseguire una applicazione e specificare parametri

argomenti



Eclipse:javadoc

- Javadoc è usato per realizzare una documentazione della applicazione fruibile all'utente
- 1. Descrizione della classe

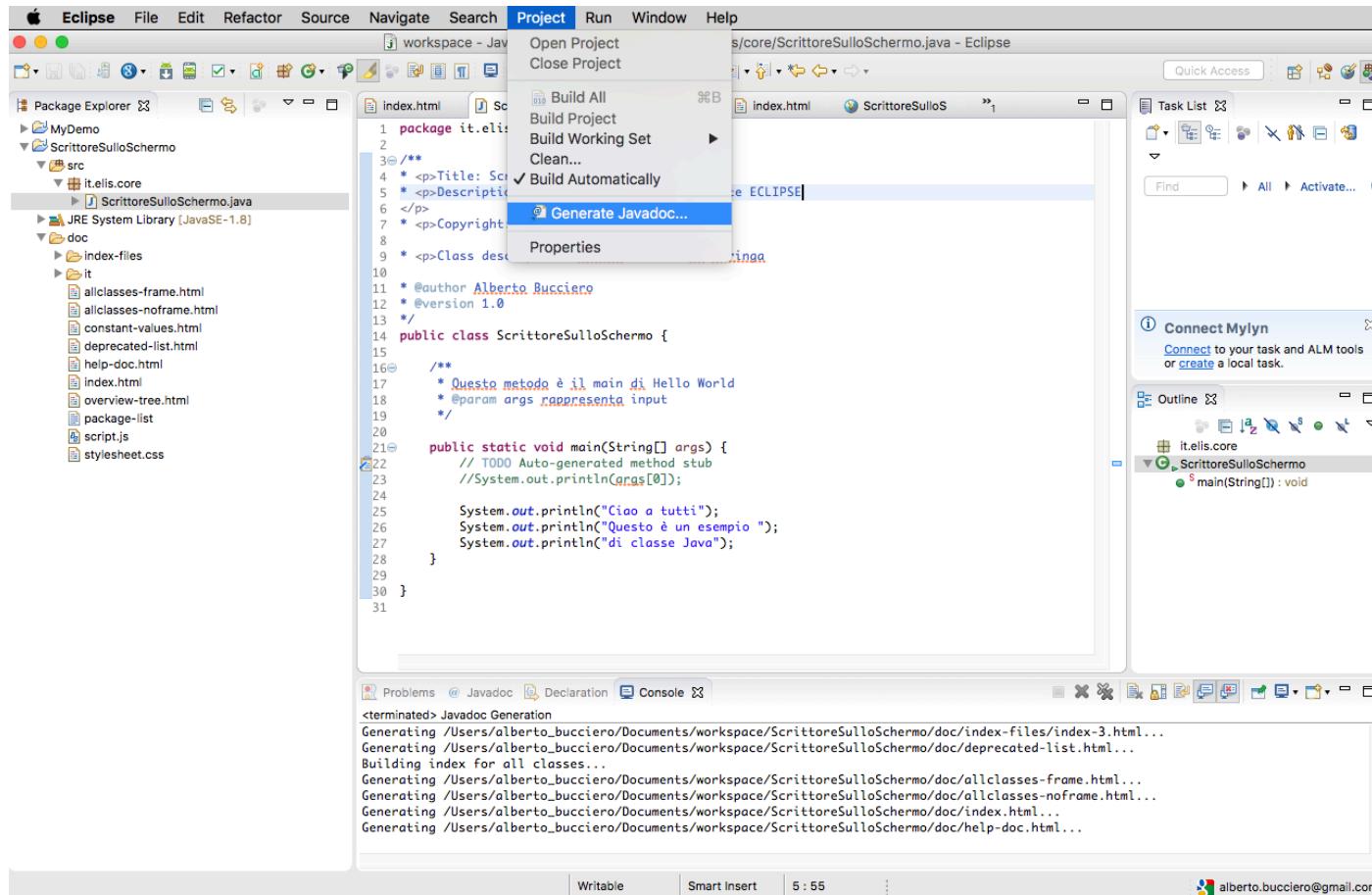
```
3④ /**
4 * <p>Title: ScrittoreSulloSchermo</p>
5 * <p>Description: esempio di uso dell'ambiente ECLIPSE
6 </p>
7 * <p>Copyright: Copyright (c) 2016</p>
8
9 * <p>Class description: stampa a video di stringa
10
11 * @author Alberto Bucciero
12 * @version 1.0
13 */
... - - - - -
```

Eclipse:javadoc

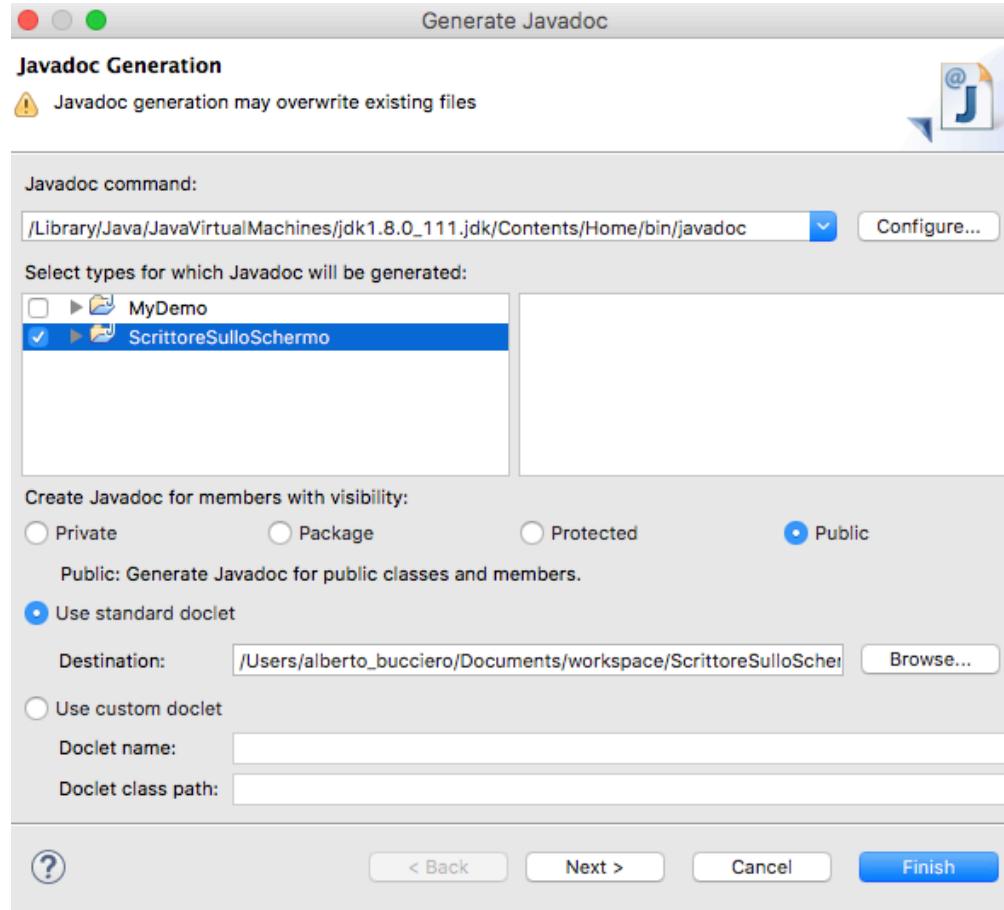
- 2. Descrizione di un metodo, allo scopo di descrivere parametri e obiettivo

```
/**  
 * Questo metodo è il main di Hello World  
 * @param args rappresenta input  
 */
```

Eclipse: generare javadoc



Eclipse: generare javadoc



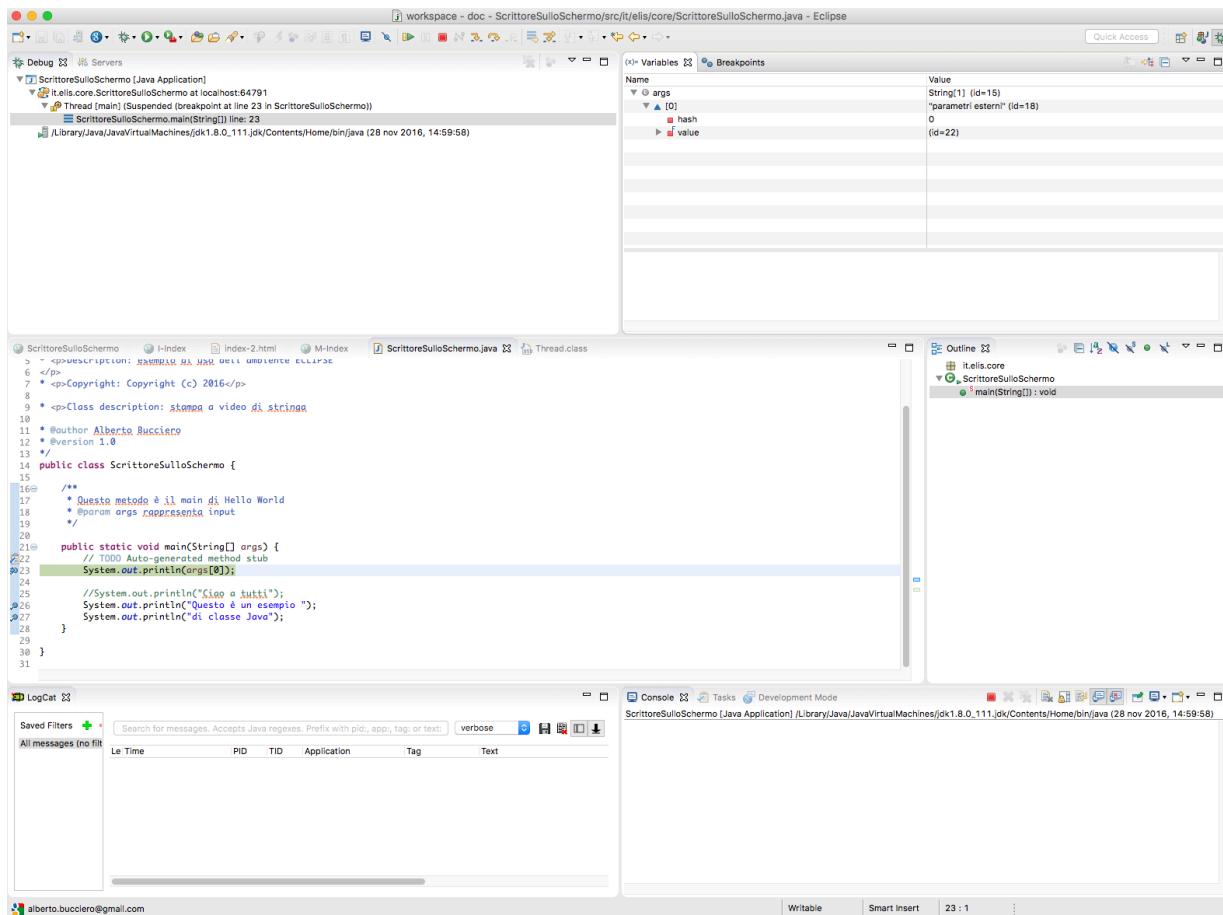
Eclipse: generare javadoc

The screenshot illustrates the process of generating JavaDoc within the Eclipse IDE. On the left, the 'File System' view shows a 'doc' folder containing several HTML files, including 'index-files' and 'it' sub-folders with various documentation pages. On the right, a web browser window displays the JavaDoc for the 'ScrittoreSulloSchermo' class. The browser title bar shows the URL: file:///Users/alberto_bucciero/Documents/workspace/ScrittoreSulloSchermo/doc/index.html. The JavaDoc interface includes tabs for 'PACKAGE', 'CLASS' (which is selected), 'USE', 'TREE', 'DEPRECATED', 'INDEX', and 'HELP'. Below the tabs are links for 'PREV CLASS', 'NEXT CLASS', 'FRAMES', and 'NO FRAMES'. Summary filters include 'SUMMARY: NESTED | FIELD | CONSTR | METHOD' and 'DETAIL: FIELD | CONSTR | METHOD'. The main content area shows the class definition: 'public class ScrittoreSulloSchermo extends java.lang.Object'. It also provides the class's title ('Title: ScrittoreSulloSchermo'), description ('Description: esempio di uso dell'ambiente ECLIPSE'), copyright ('Copyright: Copyright (c) 2016'), class description ('Class description: stampa a video di stringa'), version ('Version: 1.0'), and author ('Author: Alberto Bucciero'). A 'Constructor Summary' section lists the constructor 'ScrittoreSulloSchermo()'.

Eclipse debugging

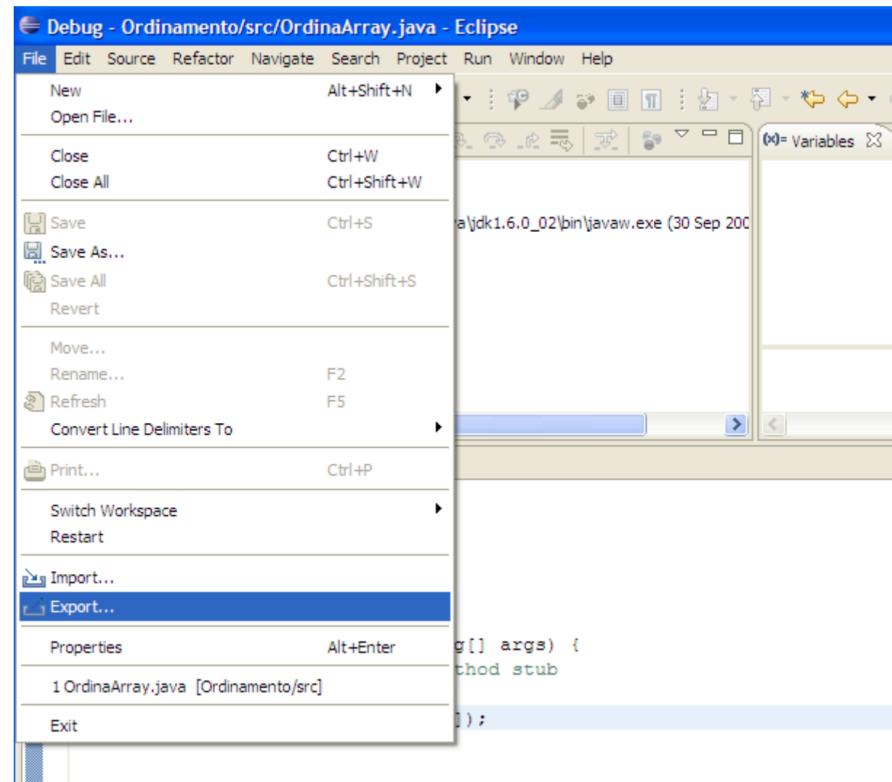
- Per inserire/rimuovere un breakpoint:
 - Selezionare il punto di interesse
 - Selezionare “Toggle Line Breakpoint” da “Run”
- Per eseguire Debug – Lanciare tramite “Debug” di “Run”
 - F5 = Step into
 - F6 = Step Over
 - Ctrl+R = per spostarsi nel punto dove è posizionato il cursore

Eclipse debugging

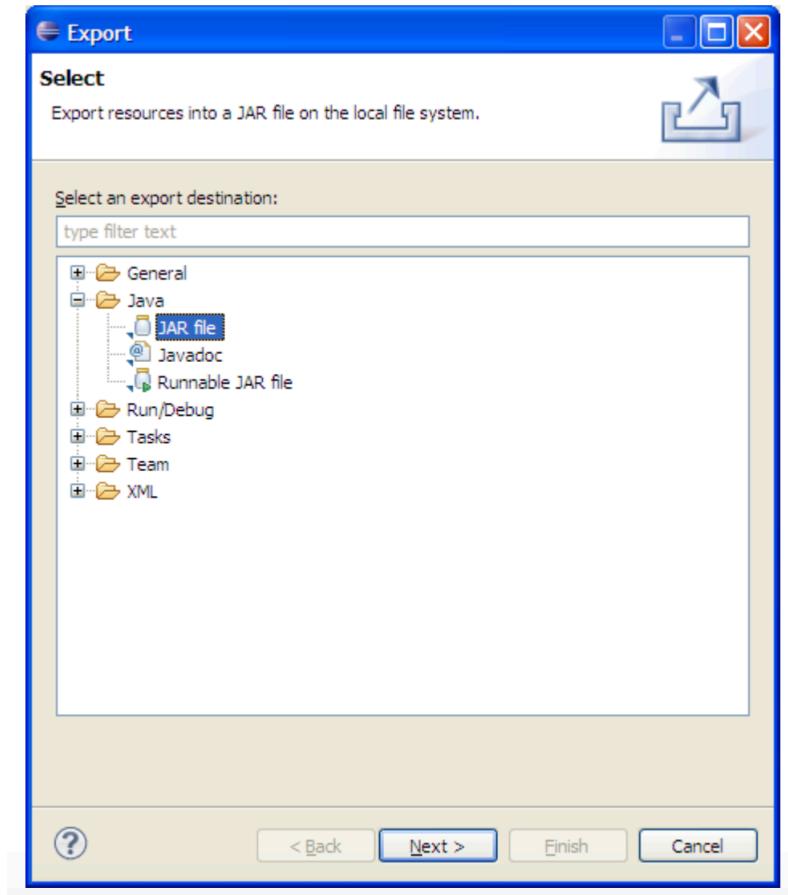


Eclipse: creare un archivio .jar

- Selezionare “Export” da menu “File”

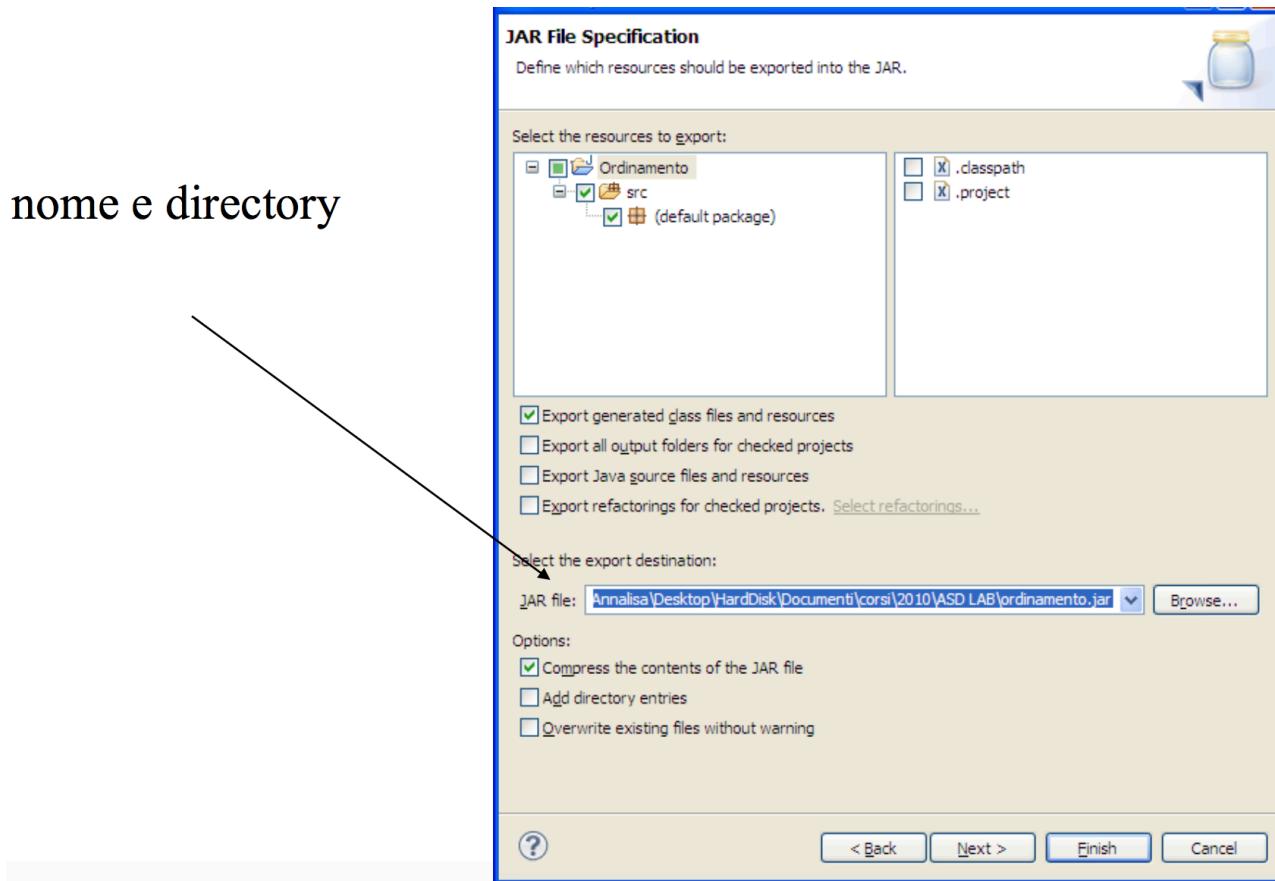


Eclipse: creare un archivio .jar



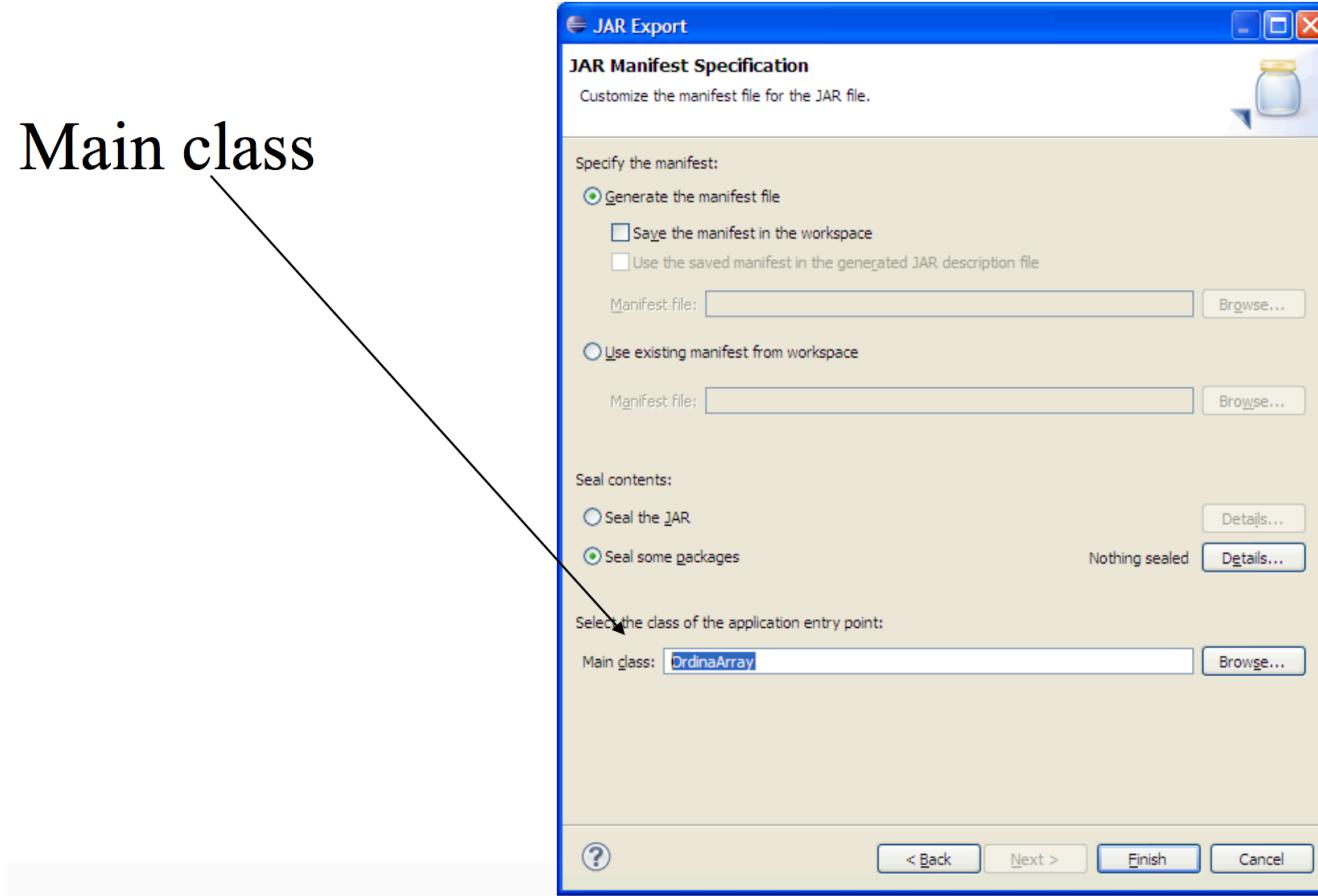
Eclipse: creare un archivio .jar

nome e directory

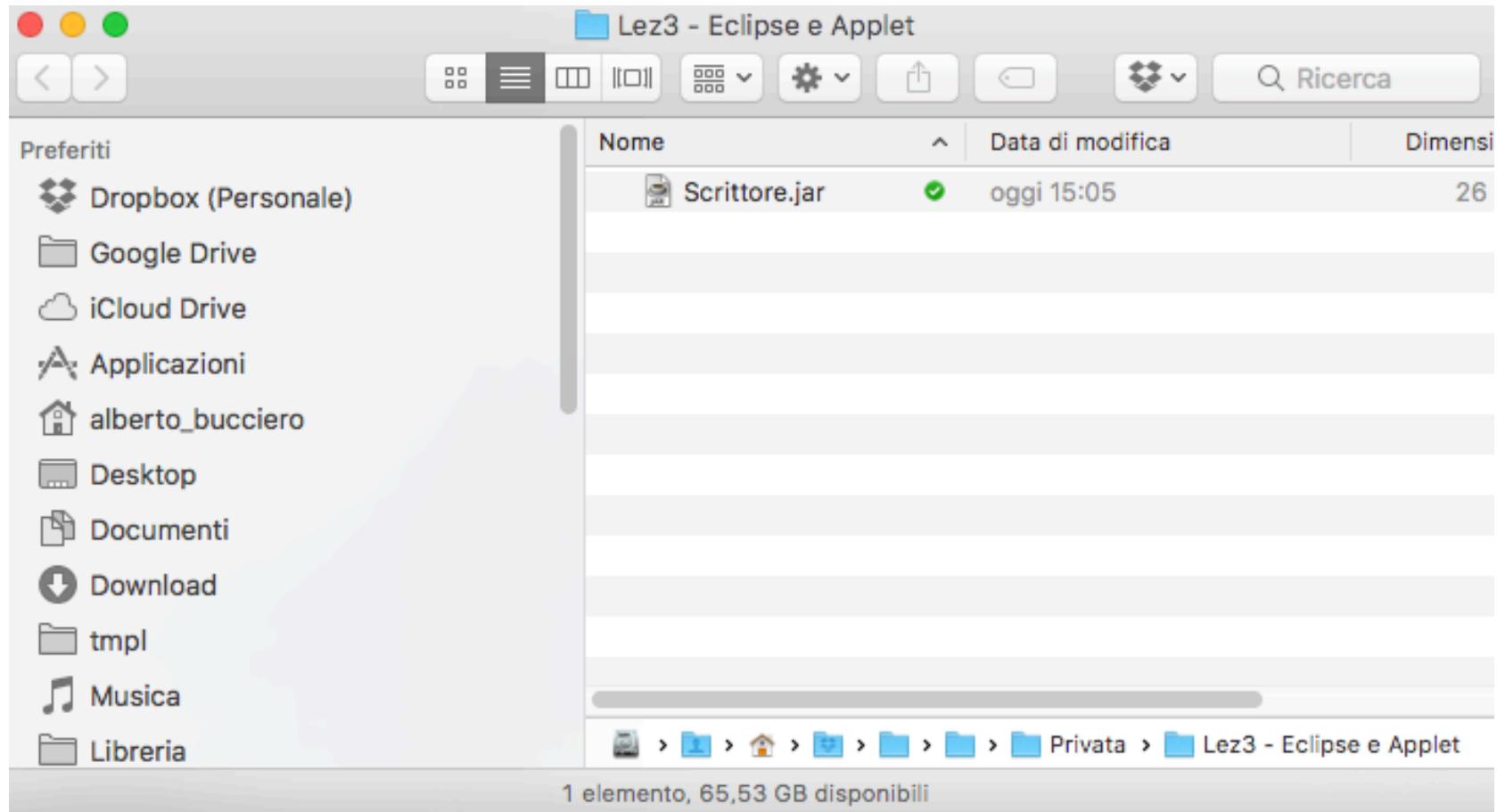


Eclipse: creare un archivio .jar

Main class



Esempio di jar



Eclipse: eseguire un archivio .jar

```
MobileSuitePRO:~ alberto_bucciero$ cd /Users/alberto_bucciero/Dropbox\ \\\(Personale\)/Corsi/Elis\ -IT\ Architect/Privata/Lez3\ -\ Eclipse\ e\ Applet/Source\ \code
MobileSuitePRO:Source code alberto_bucciero$ java -jar Scrittore.jar "prova esecuzione"
prova esecuzione
Questo è un esempio di classe Java
MobileSuitePRO:Source code alberto_bucciero$
```